AFT

REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-01-

0301

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewi ning
the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or a or
reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson D e of
Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
5/25/2000

3. REPORT TYPE AND
Final

4. TITLE AND SUBTITLE
Workshop on Semantic Spatial Databases

5. FUNDING NUMBERS
BMDO Grant F49620-98-1-0130

6. AUTHOR(S)
Naphtali Rishe, PI

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Florida International University
School of Computer Science
High Performance Database Research Center
Miami, FL 33199

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of Defense
Ballistic Missile Defense Organization (BMDO)
7100 Defense Pentagon
Washington, DC 20330-7100

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***

This grant supported a workshop on Next Generation Generation Database Design and Applications that was held at Florida International University's High Performance Database Research Center (FIU HPDRC) on April 30, 1998 and May 1, 1998. This workshop featured HPDRC-developed technology and facilitated the exchange of ideas with other researchers. Attendance was by invitation; over 70 people attended the workshop. Sessions on medical informatics, advances in database design, GIS and spatial data applications, and semantic/object-oriented database management systems were held. The workshop's keynote speakers were Professor Wesley Chu (UCLA), Mr. Richard Campanella (a Remote Sensing/GIS Specialist with the Institute for Technology Development), Professor Naphtali Rishe (FIU HPDRC), and Mr. Kent Wreder (the Corporate Director of Object Technology for Baptist Health Systems).

The workshop facilitated the professional development of graduate and undergraduate students. The HPDRC's students presented their projects at poster sessions held during this workshop and published abstracts or articles describing their work in the Workshop's proceedings.

14. SUBJECT TERMS
workshop, database design, database applications, semantic databases, spatial databases, medical informatics

15. NUMBER OF PAGES
68

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

FINAL REPORT

Workshop on Semantic Spatial Databases

F49620-98-1-0130

Florida International University
High Performance Database Research Center
School of Computer Science
University Park
Miami, FL 33199

(305) 348-1706

May, 2000

Abstract

This grant supported a workshop on Next Generation Generation Database Design and Applications that was held at Florida International University's High Performance Database Research Center (FIU HPDRC) on April 30, 1998 and May 1, 1998. This workshop featured HPDRC-developed technology and facilitated the exchange of ideas with other researchers. Attendance was by invitation; over 70 people attended the workshop. Sessions on medical informatics, advances in database design, GIS and spatial data applications, and semantic/object-oriented database management systems were held. The workshop's keynote speakers were Professor Wesley Chu (UCLA), Mr. Richard Campanella (a Remote Sensing/GIS Specialist with the Institute for Technology Development), Professor Naphtali Rishe (FIU HPDRC), and Mr. Kent Wreder (the Corporate Director of Object Technology for Baptist Health Systems).

The workshop facilitated the professional development of graduate and undergraduate students. The HPDRC's students presented their projects at poster sessions held during this workshop and published abstracts or articles describing their work in the Workshop's proceedings.

This report includes a summary of the Workshop's program, a list of attendees, and a copy of the Workshop's proceedings.

Workshop Program

Thursday, April 30, 1998

Medical Informatics

8:30 - Registration Desk Opens
9:00 - 10:00
Wesley Chu, Ph.D.
Knowledge-based
Medical Image Retrieval

10:15 - 10:30 - Break
10:30 - 11:45
Andriy Selivonenko, M.D.
Medical Informatics

11:50 - 12:30
Kent Wreder, M.S., and
K. Beznosov, M.S.
Building Information
Systems for
Healthcare Enterprises

12:30 - 1:30 - Lunch

Advances in Database Design

1:30 - 3:00
Naphtali Rishe, Ph.D.
Advances in Database
Design Methodologies

2:45 - 3:00 - Break
3:00 - 4:00
Raimund Ege, Ph.D.
Java, Object-Oriented
Databases, ODMG

4:00 - 5:00
Maxim Chekmasov, Ph.D., and
Manju Palakkat
Database Design in Oracle

4:00 - 6:00
Poster Session

Friday, May 1, 1998

GIS and Spatial Data Applications

8:30 - Registration Desk Opens
9:00 - 10:15
Richard Campanella, M.S.
Enriching Databases Through Remote Sensing
and GIS

10:15 - 10:45
David Barton, Ph.D.
Storage of spatial Data

10:45 - 11:00 - Break
11:00 - 11:30
Martha Gutierrez, M.S.
Applications of Remote Sensing Data

11:30 - 12:00
Maria Cereijo Martinez, Ph.D.
GIS and Internet Tools to Access Spatial Data

12:00 - 12:30
Elma Alvarez, M.S.
Multimedia Spatial Databases

12:30 - 1:30 - Lunch

Semantic/Object-Oriented DBMS

1:30 - 2:30
Naphtali Rishe, Ph.D.
Semantic/Object-Oriented DBMS

2:30 - 3:00
Demonstrations: Database Query Tools

3:00 - 3:15 - Break
3:15 - 5:00
Panel: Vertical Application and System Integration

4:00 - 6:00
Poster Session

Attendees

First Name	Last Name	Company
Enrique	Almendral	Florida International University
Elma	Alvarez	Florida International University
Willie	B.	Carfel Inc.
David	Barton	Florida International University
Yaman	Battikhi	Florida International University
Howard	Beck	University of Florida
Larry	Bobo	Florida International University
Angela	Bowers	Florida A&M University
David	Buker	Everglades National Park
Tarek	Chebbi	Miami-Dade County Public Schools
Maxim	Chekmasov	Florida International University
Wesley	Chu	UCLA
Debbie	Davis	Florida International University
Jorge	Dominguez	Hellmann International
Raimund	Ege	Florida International University
Bryon	Ehlmann	Florida A & M University
George	Espinosa	
Allan	Falconer	Mississippi Space Commerce Initiaive
James	Farley	University of Arkansas

First Name	Last Name	Company
David	Fay	T2C2
Bill	Feild	Florida International University
Gisela	Feild	Miami-Dade County Public Schools
Julie	Fernandez	Florida International University
Cristy	Fernandez	Miami-Dade County/ITD
Raquel	Ghersgorin	Florida International University
Hugh	Gladwin	Florida International University
Alejandro	Gonzalez	Florida International University
Scott	Graham	Florida International University
Elliot	Grossman	Mt Sinai Medical Center
Martha	Gutierrez	Florida International University
Freddy	Haayen	Florida International University
Tin	Ho	Florida International University
Andrei	Kirienko	Florida International University
Dan	Kodesh	GST, Inc.
Michael	Kon	Mt Sinal Medical Center
Venkat	Maddineni	Florida International University
Karthik	Madhyanapu	Florida International University
Shalom	Maimon	Mt Sinal Medical Center
Keith	Markbreiter	Aviation Sales Company
James	Martell	PS&R Systems, Inc.
Maria	Martinez	Florida International University
Laura	Melnik	Authorgenics, Inc.
Janlet	Minor	Aviation Sales Company
Becky	Miro	Florida International University
Khaled	Naboulsi	Florida International University
Wing	Ng II	Florida International University
Theresa	O'Connell	Florida International University
Chris	Ogier	ERADS, Inc.
Wilbis	Padron	Florida International University
Luis	Paez	Supreme International
Philippe	Pardo	Florida International University
Eduardo	Perez	Florida International University
Christian	Pesantes	Florida International University

First Name	Last Name	Company
Joseph	Pontillo	Florida International University
Guido	Pozo	Florida International University
Nagarajan	Prabhakaran	Florida International University
Maria	Puentes	Hellmann International
Jim	Puglise	Information Mgmt Syst, Inc.
Amado	Ramirez	Supreme International
Naphatali	Rishe	Florida International University
Andriy	Selivonenko	Florida International University
John	Simmons	Carfel Inc.
Roger	Smeds	Authorgenics, Inc.
Sangeetha	Sridhar	Florida International University
Darrell	Tidwell	Everglades National Park
Jinny	Uppal	Florida International University
Frank	Urban	Florida International University
Juan	Valiente	University of Florida
Alexander	Vaschillo	Florida International University
Dmitry	Vasilevsky	Florida International University
Kent	Wreder	Baptist Hospital
Xiangyu (Grace)	Ye	Florida International University
Baolin	Yin	Florida International University

WORKSHOP ON NEXT GENERATION DATABASE DESIGN AND APPLICATIONS

EDITORS: Naphtali Rishe and Maria Martinez

MULTIMEDIA SPATIAL DATABASES*

Elma L. Alvarez, and Naphtali. D. Rishe

High Performance Database Research Center
School of Computer Science, Florida International University
University Park, Miami, Fl 33199
(305)348-1706 Fax: (305)348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

Multimedia has become an increasingly important component in business, industry, science, entertainment and education. Its primary role has been the improvement of performance in all these fields. This paper will describe the design and implementation of a multimedia spatial database system. Using the Semantic Object Oriented Database Management System that has been developed at the High Performance Database Research Center (HPDRC) at Florida International University (FIU), semantic databases have been designed, and created. Textual data including satellite, instrument, date, latitudes and longitudes, remote sensed and digital data (i.e., Landsat TM (Thematic Mapper) and Digital Aerial Photography) and multimedia data have been loaded. A multimedia introductory sequence has been developed together with a main interactive flight and color composite application to form an edutainment Windows based CD-ROM. This software combines real entertainment with interesting science to provide a versatile and practical system for users from different technical backgrounds.

INTRODUCTION

Over the past few years, the use and availability of remote sensed data has increased exponentially. At the same, time multimedia data is also increasing and becoming a requirement in spatial information system. Some communities are already exploring the possibilities of integrating multimedia data and spatial data together into one system. The High Performance Database Research Center (HPDRC) at Florida International University is one of these research centers.

At HPDRC, in addition to several megabytes of multimedia data, we have a wide variety of spatial data sets from several sources including Landsat TM data deployed by NASA's Goddard Space Flight Center and Digital Aerial Photography acquired from the USGS. Due to the large amount of data inherent in these types of data products, we found a need for a computer-based system able to efficiently store, manipulate, analyze and display this information. Hence, a multimedia spatial database system with an interactive graphical interface was the solution.

BACKGROUND

What is Multimedia?

Multimedia is a tight integration of several computer technologies including text, audio effects, such as sound effect, and music, video, 2D/3D graphics, animation, interactive programming and a great design. Today, businesses are using computers with multimedia capabilities to train employees, present new

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

products to clients, and in any aspect that increase the productivity of the organization. Multimedia is promoted as the technology for the future from the perspective of applications and its role in improving business performance. One of the largest and fastest growing uses of multimedia is the use of the Internet. A growing number of businesses, however, are looking to multimedia for real-life business solutions because of its capacity to integrate text, drawings, full-vector graphics and full-motion video.

What is a Multimedia Spatial Database?

A multimedia spatial database is a database that contains spatial data (i.e., digital and remote sensed satellite data) and multimedia data (i.e., sound, video, music, graphics, digital movies, animation, and images), in addition to the textual and conventional data.

Semantic Object Oriented Database Management System

Under NASA sponsorship, the HPDRC has developed a High Performance Semantic Database Management System (DBMS). This semantic DBMS has been developed with an object-oriented approach and is based on the Semantic Binary Model. Thus it satisfies the three essential needs of many database applications: strong semantics embedded in the database to handle the complexity of the information, storage of multi-dimensional spatial, images, scientific and other non-conventional data, and very high performance that allows rapid flow of massive amounts of data. The semantic parallel architecture of this database system provides efficient and flexible access to a large collection of data stored on various physical devices. Further, data reference transparency is an inherent property of the semantic binary model system (Rishe, 1992a) (Rishe, 1994).

The Semantic Database model is potentially more efficient than conventional models for two main reasons. First, all the physical aspects of the representation of the information are invisible to the users. This additionally creates a potential for optimization by allowing more changes to the database without affecting the users' programs. Second, the semantic system knows more about the meaning of the user's data and about meaningful connections between such data. This allows that knowledge to be used to organize the data in such as way that meaningful operation can be performed faster. (Rishe, 1992b).

Most of these features are requirement in a DBMS system (Adjeroh, 1997) to be able to efficiently support a multimedia spatial database. Since this type of database has to support the storage of multi-dimensional spatial data, multimedia data and textual data all in one database. Besides, it needs to have a high performance engine to allow massive and efficient retrieval of all these data types.

APPLICATIONS

Multimedia computing is expanding quickly into a number of new application areas, such as video conferencing, educational, entertainment, marketing, business, Internet, science and edutainment. Today, businesses are using computers with multimedia capabilities to train employees, present new products to clients, advertise existing and new products, and for many other real-life business solutions. In the entertainment field, the demand for the use of multimedia has increased such that most of the interactive games and videos are multimedia produced. Science is another major field where multimedia is increasingly considered a requirement in their systems. Multimedia data has been incorporated into Geographic Information System (Kraak, 1996) and into spatial information system. Multimedia also plays a main role in the edutainment field. This paper will describe in detail the design and implementation of TERRAFLY, an edutainment application that use a multimedia spatial database.

COMPONENTS

In the development of a multimedia spatial database and its user interface, three major components: database, software and hardware must be tightly integrated. The selection of these three key components makes the difference in the development and performance of the system. For the TERRAFLY system the following components were used:

- Database: a multimedia spatial database was built using a Semantic Object Oriented Database Management System. The database currently stores textual, spatial and multimedia data.
- Software: the Multimedia Authoring Tool used is Macromedia Director 6, and the programming languages used are Visual C++ and Lingo.
- Hardware: microphone, speakers, video card, sound card, digital camera, scanner, and others.

TERRAFLY MULTIMEDIA SPATIAL DATABASES

The multimedia spatial database used for TERRAFLY was developed using the semantic/object oriented model approach. A Semantic Database schema is a set of categories, relations, and database types. A category is a specification for database abstract objects that belong to that category. Each category may have several relations with other categories and data types. A relation from a category to a data type is called an attribute, and a relation from category to a category is called an abstract relation (Rishe, 1992a).

The databases used for this application have to store the spatial, multimedia and semantic data together in the same database. This is a feature that most DBMS systems lack. They store the spatial data separately from the textual data (Waugh, 1987) which makes the system inefficient and difficult to use (Rishe, 1994). TERRAFLY currently has two databases: Landsat database and Aerial Photography (DOQ, Digital Orthophoto Quad) Database.

Landsat database

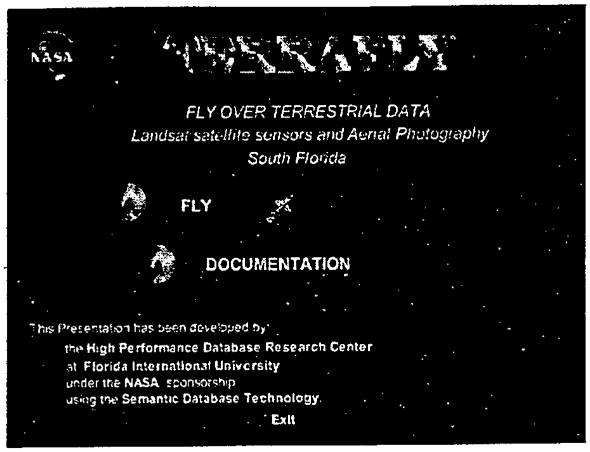
This database needs to store textual information about the Landsat data including: date, satellite, path, row, sensor number, latitude, longitude, and spatial data including one quad of satellite data covering an area of 2850 square miles and corresponding to the Miami-Dade County region of the state of Florida. In addition it needs to store some multimedia data including sound data, voice data, and pictures. The spatial data was divided into tiles of 160 x 312 bytes and then compressed using g-zip algorithms before storing on the database.

Aerial Photography database

This database, as with the Landsat database, needs to store textual data about the digital Aerial Photography (DOQ) including: latitude, longitude, date, rows, columns, and digital data including about 70 quad of aerial photography data covering an area of 1400 square miles over the Miami-Dade county area. In addition to the textual and spatial data this database need to store multimedia data including voice data, sound and topography data. Users can make a query to the database requesting a geographical location's identification and the response will be in the form of a voice from the computer describing the location/place and giving the corresponding address.

TERRAFLY IMPLEMENTATION

This system is completed independent from the data upon which it operates. It retrieves all necessary information (data) from the semantic database, making this interactive system a generic one that works for a variety of data sets. In addition to the databases, this system consists of two major parts: introductory part developed used Macromedia Director 6 and the main system part developed using Visual C++. Figure 1 shows a snapshot of the introductory sequence. This frame guide the users to link to external Internet Web Pages to peruse more detail information about the HPDRC, FIU and our sponsor, NASA. They may also review documentation about Thematic Mapper data from the Landsat 5 series of satellite. And they can take a flight by launching the interactive flying application. This is the most important part of the system and it has been divided into sections for more detail explanation.



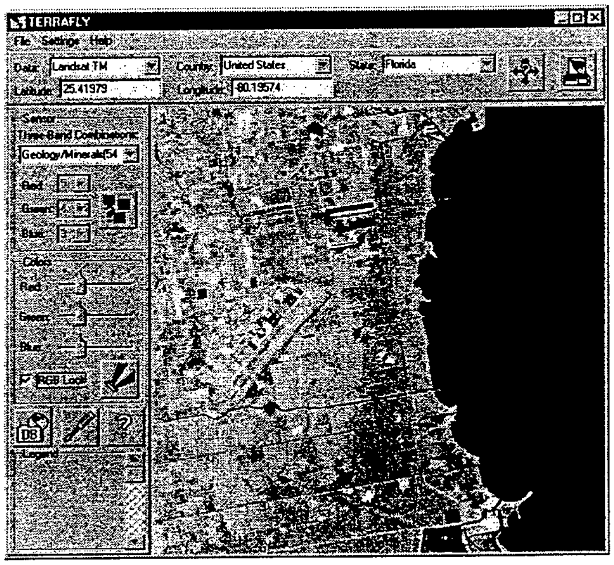
FLY OVER TERRESTRIAL DATA
Landsat satellite sensors and Aerial Photography
South Florida

This Presentation has been developed by
the High Performance Database Research Center
at Florida International University
under the NASA sponsorship
using the Semantic Database Technology.

Figure 1: TERRAFLY Introductory sequences

User Interface

We developed a friendly graphical user interface that is simple to use while maintaining a high level of flexibility and advanced data manipulation techniques. The main features of the interface are a top bar, drop-down menus, buttons, and text-boxes. These allow users to easily manipulate the data, and retrieve all the needed information. Some of the menu options include: data set, state, latitude, longitude, print, help, customized 3-sensor (bands) combinations, advanced 3-band color composite, RGB insensitive control, and main flying windows. Figure 2 give a snapshot of this interface and the functionally of each option are described as follow:



Data: Landsat TM
Country: United States
State: Florida
Latitude: 25.41979
Longitude: -80.19574
Three-Band Combinations: Geology/Minerals(54
☑ RGB Lock

Figure 2: TERRAFLY Interface

- Data set: This option allows the users to select from the different data sets currently stored in the database. There are two spatial data sets: Landsat and Digital Aerial Photography. Figure 3 gives an example of these two data sets:
- State: currently there is only one state: Florida, but additional states and regions will be added to the database
- Latitude & Longitude: These two text-boxes provide the users with the Latitude and the Longitude for the center point of the image currently been display. Users can enter a Latitude and Longitude and query button the database to retrieve the image corresponding to this geographical location.

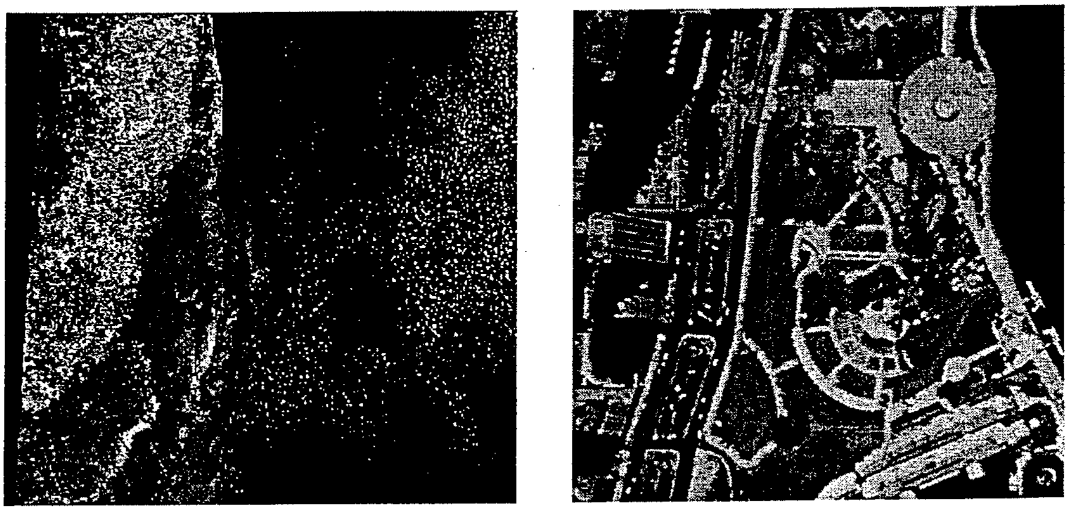


Figure 3: Sample of Landsat data (left) and Aerial Photography (right)

- Print: This option allows the users to print some areas of the displayed spatial data.
- Customize three bands (sensor) combination: Users can select some predefined and useful three-sensor combinations to view false color images from a drop-down menu. Every time that a new combination is selected, a different set of images/bands is retrieved from the database and the selected false color image are computed and displayed within the window
- Advanced three-band color composite: For the more scientific users, this application allows entrance to any three-band combinations (RGB) (e.g. 654) that the user is interested in studying or analyzing. After the combination is entered and the query button is pressed, the data is retrieved from the database, the image is computed and then displayed. For the Landsat the users are allowed to select from a list of 7 possible sensors, but from the Aerial Photography there is only three possible sensors.
- RGB intensity control: This option allows the users to increase or reduce the intensity of any of the bands that represents the colors (Red, Green or Blue)
- Main Flight window: This is the main window that displays the spatial data image and that allows the users to fly over the data at different speeds and directions by positioning the mouse within the image.

Display Process

This system displays the spatial data images in static and dynamic modes. This section is going to concentrate on the static displaying process. The next section (*Data Animation)* will discuss the dynamic displaying process and analysis.

During the static display process, a data file for one particular date is retrieved from the database and placed in a buffer. This process includes the retrieval of three blocks of data from the database, where (x, y, and z are numbers from 1-7): one for band x, one from band y and one from band z. Then data decompression and creation of the new false coloring (color composite picture) is preformed on the data. Once this process is completed the new data in BIP format is display on the screen using window functions.

Data Animation

During the data animation process, more computation and data access needs to be performed in order to achieve a dynamic view of the data. At the same, time high efficiency algorithms need to be running to cope with the massive amount of data requested by the users and the system to produce a smooth flight.

The first step to achieve the animation is the same as previously described on the *Display Process* section. After this, as the user flies over the data, new blocks of data are retrieved from the database and placed on a circular buffer to reduce the memory space require to run the application. Once this is completed, the display function is activated and the new data is displayed on the screen.

Data Retrieval

This application is constantly retrieving information from the semantic database. Every time that the user positions the mouse over the image to fly over it, the application needs to perform a massive retrieval of three times the amount of data displayed on the screen. This large amount of data is necessary for each final image because data from three sensors needs to be retrieved from the database and processed before being displayed. Further, when a new three-band color combination is requested, a query needs to be performed on the database to retrieve the three corresponding sensors' data.

DATA

Semantic Data

Several megabytes of semantic, spatial and multimedia data are been used in this application. As was stated previously, this system does not merely display spatial data. It provides users with information about the data including latitude, longitude, path, row, and so forth. In order to achieve this, all the textual information is combined with the spatial and multimedia data in the same database. This is needed to better meet all requests coming from the users and allows them to view and receive information about the spatial data at the same time that they are looking at the images

Spatial Data

The spatial data currently store in the database is mainly in a raster format (Muffin, 1987). It includes:

- One quad of Landsat (Thematic Mapper) TM data. This data is 30 meters resolution and covers an area of 2850 square miles belonging to the Miami-Dade county area. Figure 3 provides a sample of the Landsat data used in this system.
- Digital Orthophoto Quad (DOQ): 72 quads of Digital Aerial Photography covering an area of approximately 1400 square miles. This data dates from 1994 and 1995, and has a resolution of 1 meter. Each data file (quad) is 150MB, for a total of approximately 12GB of digital data. Due to the limited capacity of the current CD-ROMs this data was compressed using JPG compression, so it could fit into one CD.

Multimedia Data

Sound data and pictures have been loaded into the database. We are in the process of loading voice/text data files that contains a description of the spatial data on the screen; in addition to the names and addresses of main sites like Florida International University, Miami International Airport and many other main places in the Miami-Dade County area.

Compression and Storage

Due to the large amount of spatial data (12GB) used by this system, in addition to the multimedia and textual data, and having the final output device CD-ROM, several compression algorithms were explored and used. The two methods used were JPG compression and g-zip, a compression program based on the Lempel-Ziv algorithm (Ziv, 1997). Minor changes were made to the original g-zip programs to write the output to memory instead of a file and to port it to the Windows environment. In this way, we have memory to memory compression that is useful to get the data file compressed, put it in memory and then write it to the database or display it on the screen. The spatial data was then stored compressed in the database and decompressed on the fly as requests to the database were performed.

APPLICATION RESULT

The resulting system has some major advantages including:

- Portable: The resulting Windows-based software is packaged into a CD-ROM that contains the semantic database, the introductory sequence and the main interactive fly and image composite application. Thus, it can be ported from one computer to another one very easily without the need of any additional files.

- Spatial Information System: This system has some spatial information system features:
  - Capability to fly over Landsat TM data and DOQ (Digital Aerial Photography) data at different speeds and directions by positioning the mouse within the image.
  - This system has facilitated the manipulation, study, analysis and interpretation of remote sensed data including Landsat and Digital Aerial Photography.
  - Customized three-sensor (band) combinations: the user can select some predefined and useful three-sensor combinations to view false color images from a drop-down menu.
  - Three-band combinations (RGB): For the more scientific users, this application allows the users to enter any three band combinations, RGB, (e.g. 654) that the user is interested in studying or analyzing.
- Educational & Entertainment: It provides entertainment for people looking for a game and at the same time teaches students and technical people looking for information about remote sensing and color composite images. Using this system users can get detailed information on what Landsat's bands (sensors) need to be combined to get the results they are seeking based on the wavelength of each sensors.
- Generic System: This system is completely independent from the data upon which it operates. It retrieves all necessary information (data) from the semantic database. Therefore, new data sets can be loaded into the database without a need to change the main system.
- Efficient: The Semantic Object Oriented DBMS provides efficient simultaneous retrieval of massive amounts of data to multiple users, and ensures better logical properties such as a comprehensive enforcement of integrity constraints, greater flexibility, and substantially shorter application programs (Rishe, 1992a).
- Data geo-location: All the displayed satellite images are geo-located by giving the latitude and longitude of the center point of the image. We are in the process of adding the latitude and longitude for every point on the screen.
- Secure data: Data is secure for two reasons. First, the databases are used and distributed in a CD-ROM, so no changes can be performed on the data. Second, the data is stored in a semantic database, which provides protection and security for the data and at the same time, enforces consistency of the stored data.

CONCLUSION

The design and implementation of this interactive multimedia spatial database system covered several areas. First, its Semantic Database was designed and created, and all the textual, multimedia and spatial data has been loaded into the database. Then, the main areas of the system – the user interface, displaying process, data retrieval and data animation - were implemented with the help of some tools and devices, at the same time that the introductory multimedia sequence was developed. Finally, all these components were tightly integrated to form a versatile and practical multimedia spatial system.

The resulting system has facilitated the storage, manipulation, analysis and display of the spatial and digital data. At the same time, it has offered an entertainment device for non-scientific people. Although this system still under development, it has already demonstrated how the incorporation of multimedia data into a spatial information system can offer users a powerful new interactive visualization environment that will satisfy their needs. This multimedia spatial database system allows a better exploration and analysis of remote sensed data by the integration of multi-data sets.

REFERENCE

Adjeoh, D., Nwosu, K., “Multimedia Database Management –Requirements and Issues”, Vol. 4, No.3 July-September 1997, pp.24-33.

Kraak, Meeno-Jan, “Integrating Multimedia in Geographic Information Systems”, Vol. 3 No.2, IEEE Multimedia, summer 1996, pp.59-65.

Muffin, G., “Raster versus Vector Data Encoding and Handling: A commentary” Photogrammetric Engineering and remote sensing, Vol.53, No. 10, 1987, pp. 1397-1398.

Rishe, N., “A Database Design: The Semantic Modeling Approach. McGraw-Hill, 1992, pp.528.

Rishe, N., Li, Q., “Storage of spatial data in Semantic Databases”, Computer in Engineering 1994, (Preceding of the 1994 ASME International Computer in Engineering Conference, Minneapolis, MN, Sept 11-14, 1994) ASME, pp. 793-800.

Rishe, N., Accepted grants proposal to NASA. Grant NAGW-4080. 1992.

Waugh, T. C., Healey, R. G., “The GEOVIEW design: A relational database approach to geographical data handling”, International Journal of Geographical Information System, Vol. 1 No. 2 1987, pp. 101-118.

Ziv, J. and Lempel, A. A Universal Algorithm for sequential Data Compression, IEEE transactions of Information Theory, Vol.IF23, No. 3, May 1977.

Using OODBMS to Create an Agricultural and Natural Resources Knowledge Base

Howard Beck

IFAS Information Technologies
PO BOX 110495
University of Florida
Gainesville, FL 32611
hwb@agen.ufl.edu

As with many large organizations, the Institute of Food and Agricultural Sciences (IFAS) at the University of Florida is involved in managing a large and diverse amount of heterogeneous information from many experts in many different but interrelated fields of study. Since the 1980's, efforts have been underway to build a digital library of IFAS information to help serve its teaching, research, and extension needs. Semantic data models and object-oriented database management systems (OODBMS) have provided the theoretical basis for constructing this digital library. A semantic data model called Candide was developed as a research project and implemented as an OODBMS. It has been applied to a wide range of problems in modeling agricultural and natural resource knowledge. Numerous projects making use of this technology are briefly described below.

Candide Semantic Data Model

Candide (Beck, Gala and Navathe, 1989) was developed as a way of applying the work being done on semantic networks in artificial intelligence to database management. In particular, Candide is an adaptation of the KL-ONE family of semantic networks. Candide is intentionally designed to contain a few simple constructs. It supports formal definitions of classes, instances, attributes, and attribute restrictions. Attributes can have one or more values of type class, instance, integer, string, floating-point, range, set, ordered set, and composite (an imbedded object). Candide does not support methods. The lack of methods and the simplicity of the model were designs to permit operations on database objects that would not be possible if complex constructs or methods were included. In particular, the KL-ONE family exploits the notion of classification by which new classes and instances are automatically classified within the existing class taxonomy by virtue of the attribute restrictions. A new class or instance is classified below the most specific class for which it can satisfy the attribute restrictions specified for that class. Classification can be exploited as a query processing technique by formulating the query as a Candide class, classifying this query class, and then the location within the taxonomy where the new class is placed by the classifier points to the results of the query. Additional work has been done in conceptual clustering by which new classes in Candide can be induced by comparing two or more instances and abstracting their common attributes and values (Beck, Anwar and Navathe, 1994).

Candide has been implemented as a research project. Although it is a data model with interesting properties, it has limitations in storage management and is not suitable for large scale database management. Currently Candide is being implemented on top of a commercial object-oriented database management system in order to overcome this limitation. Nevertheless, Candide has been successfully applied to many areas, as illustrated by the examples below, and has demonstrated the importance of semantic data models in a diversity of applications.

Florida Agricultural Information Retrieval System (FAIRS)

Candide was originally developed as part of a project on information retrieval. FAIRS was created in the early 1980's as one of the first electronic information systems for agriculture. FAIRS currently includes a collection of 3000 agricultural extension publications along with thousands of images of crops, plants, and pests. It is in use by agricultural agents throughout Florida. FAIRS has evolved over the year, along with technology. It was distributed on CD-ROM in the early 1990's, and was moved to the Web in 1994 (http://hammock.ifas.ufl.edu). The web site currently receives 100,000 visitors (averaging a total of 1,000,000 hits) each month.

In FAIRS, all documents and related information are represented as data objects using a Document Object Model (DOM). The FAIRS web server consists of the Candide OODBMS, and HTML files are generated from the data objects as needed in real time. Although SGML and HTML have been important factors in the development of this database, the data objects have emerged as the most effective way of representing document structure (Williams and Beck, 1996). The objects also provide a high level of abstraction. This has enabled the FAIRS database to survive many technical transitions and changes in format over the years. The semantic data model also provides an advantage over conventional tagging systems such as HTML because the data objects are able to represent concepts and content information as well as the structure of documents. In the next phase of this project, tagging languages are being totally rejected as a way of representing or even transmitting documents. Instead, this will be done entirely with data objects.

Knowledge Acquisition Editors using Java/CORBA/OODBMS

With the advent of Java, CORBA, and commercial OODBMS systems on the Internet, there is a natural environment for extending this work. Currently a series of specialized input forms and editors is being developed in Java to help experts in IFAS enter their knowledge into the database. ObjectStore, Poet, and Versant are being evaluated as commercial platforms for a distributed object database which will run statewide, linking growers, county extension agents, and agricultural experts to this knowledge base. CORBA is used to transmit objects between the client and the database.

For example, an editor complete with all the traditional functions of a word processor can be built using Java components. Authors can use this tool directly on the Web to create and edit the publication database in FAIRS. But unlike traditional word processors, the editor works directly on database objects, and all content is stored on the Web server OODBMS. By using the Internet, these documents can easily be shared among authors and reviewers, and the OODBMS can also facilitate this collaborative work environment. There are no word processing files involved. There is no tagging involved. Previously SGML had been used to assist in converting tagged word processing files to the data objects. Now this can be done directly and updates can appear on-line immediately. The editor contains highly structured templates for specifying items such as title, author, and section headings. Using SGML, this was an error-prone process, but the templates in the new editor simplify the process for authors while reducing errors.

Pesticide Information Database

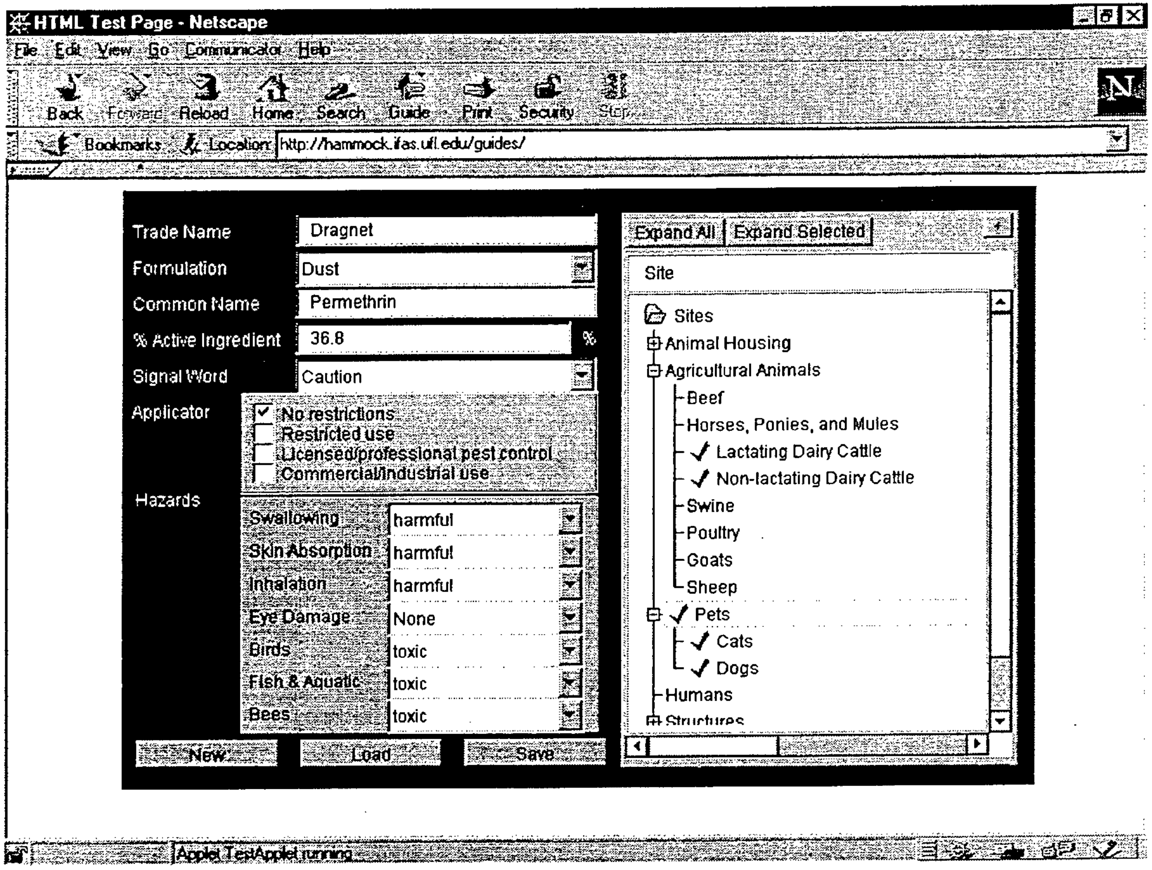
Much of agriculture involves pest control, and the pesticide product labels which need to communicate information on proper use and safety are extremely complex and without common format. This precludes the use of relational databases for storing pesticide information, whereas the complexity of these labels can be handled adequately using objects. Currently a project involving a large set of documents on pest management is being implemented using specialized editors (Figure 1) such as described in the previous section. Pesticide labels contain a good deal of taxonomically oriented information. For example, a pesticide may be used on a particular site such as Urban->Household->Indoors->Food Area, and special restrictions apply depending on the site. The inheritance features of the OODBMS facilitate queries which must operate on this taxonomy.

Decision Information Systems for Citrus

Other applications include decision support systems (expert systems, statistical models, computer simulations of crops and other ecosystems) such as the DISC project (Decision Information Systems for Citrus). In DISC, the data model is used to describe a wide variety of citrus production practices, variety information, and statistical models of disease occurrence, tree growth and yield. During the development of this project, citrus experts are using the Java/CORBA/OODBMS software to build decision models. Since the project participants are geographically distributed at various research centers throughout Florida, the Internet is very useful as a tool for knowledge acquisition.. For example, a decision matrix is used for relating citrus varieties to various environmental conditions such as soil, climate, water availability, and other factors. Experts at the Lake Alfred Citrus Research and Education Center can edit the decision matrix on-line in collaboration with other experts and a programming team located in Gainesville. Eventually, growers throughout the state will be able to participate in this collaboration as well, where all data, observations, decision rules, and other information are collected in the database.

Florida Automated Weather Network (FAWN)

FAWN was constructed in 1997 by IFAS to provide real-time agricultural weather information to Florida growers. FAWN currently consists of 16 automated weather stations located in rural areas ranging from Gainesville to Homestead. Each station gathers an array of data including 4 different temperature readings, wind speed, solar radiation, rainfall, and other parameters. The Internet is used to gather this information every 15 minutes, where it is stored in a relational database located on a Web server in Gainesville. A web site (http://fawn.ifas.ufl.edu), featuring a Java applet which can be used to plot various parameters over time, provides a way for growers to access this information. Although FAWN uses relational rather than OODBMS technology, it illustrates that when data are naturally suited to a tabular format, relational databases can be used in conjunction with OODBMS.


HTML Test Page - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Location:
New Load Save
Expand All Expand Selected
Applet TestApplet running

Location: http://hammock.ifas.ufl.edu/guides/

Trade Name	Dragnet
Formulation	Dust
Common Name	Permethrin
% Active Ingredient	36.8 %
Signal Word	Caution
Applicator	☑ No restrictions □ Restricted use □ Licensed/professional pest control □ Commercial/industrial use
Hazards	
Swallowing	harmful
Skin Absorption	harmful
Inhalation	harmful
Eye Damage	None
Birds	toxic
Fish & Aquatic	toxic
Bees	toxic

Site

- Sites
  - Animal Housing
  - Agricultural Animals
    - Beef
    - Horses, Ponies, and Mules
    - ✓ Lactating Dairy Cattle
    - ✓ Non-lactating Dairy Cattle
    - Swine
    - Poultry
    - Goats
    - Sheep
  - ✓ Pets
    - ✓ Cats
    - ✓ Dogs
  - Humans
  - Structures

Figure 1. Pesticide label data entry form using Java, CORBA, and OODBMS

Environmental Education Multimedia Project

Multimedia applications also integrate nicely within the OODBMS. All elements of the multimedia application, including layout, highly customized devices, and interactive applications as well as the usual images, sounds, video clips, and animations, can be represented and stored in the OODBMS. This contrasts with traditional multimedia authoring tools such as Macromedia Director which uses proprietary file formats to captures this information. A strong advantage of the OODBMS approach is that the multimedia application integrates directly with other applications and data. The OODBMS also promotes reuse of multimedia components. This approach has been used in an environmental education project featuring a variety of multimedia and including virtual reality tours of natural ecosystems in north Florida.

Image Archive for Pest Identification using Digital Cameras

Recently a project involving archiving photographs of pests (insects, weeds, and diseases) taken in the field using digital cameras is being built using commercial OODBMS tools. The photographs will be used instead of mailing plant materials or insect specimens to specialists for identification. Agricultural extension agents take photographs of pests observed in the field. A Java applet is used to enter the photographs and other information which will support identification of the pest. The Internet is used to transmit this information among pest identification experts and agricultural workers in the field.

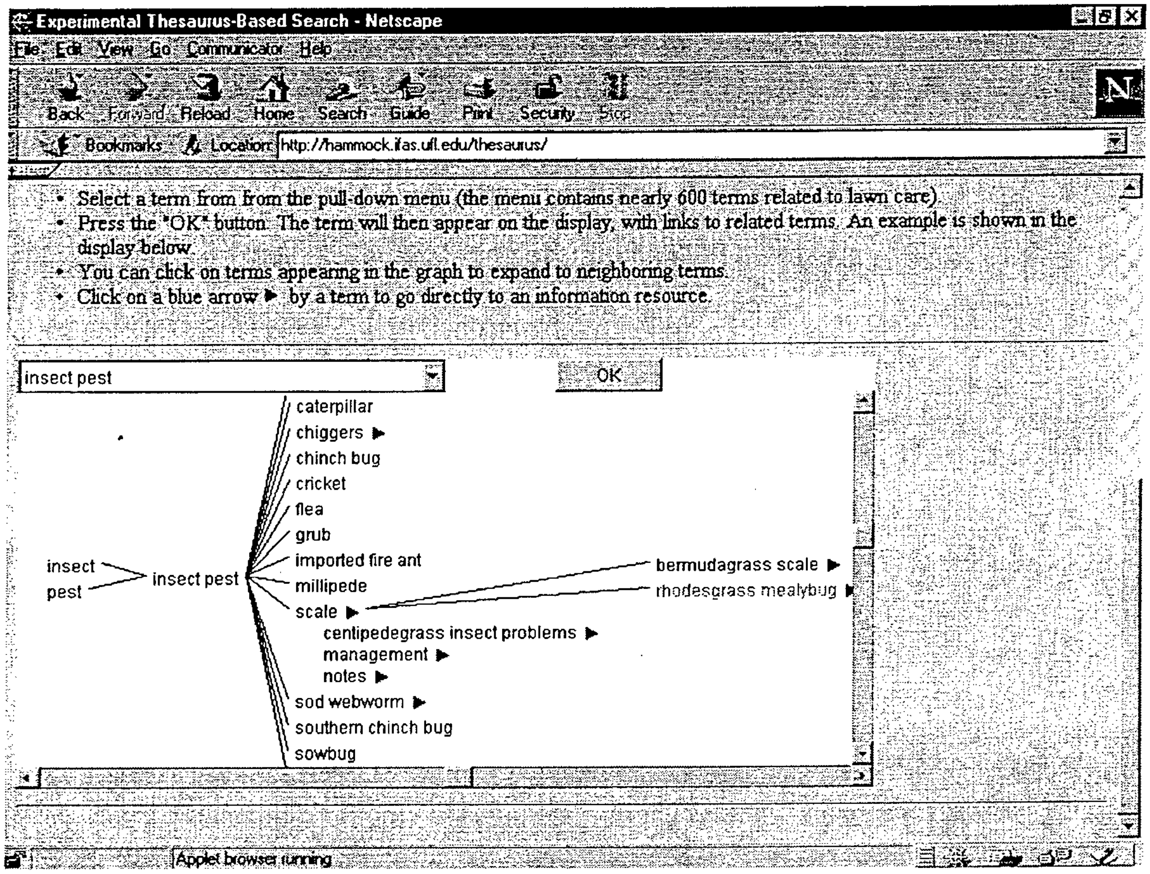
Although this can be done using e-mail by sending the photographs as attachments, that approach fails to capture important information into an archive where it can be saved, analyzed, and reused. For example, there is currently a shortage of pest images which could be used as illustrations in publications, in workshops, or in pest identification programs. Previous attempts at building expert systems to help identify pests have suffered from lack of photographs (there are no good photographs available for many pests), and because a single pest may appear differently in different situations (there are differences in hosts, life stages, or even lighting conditions). By systematically storing the photographs being captured using these digital cameras, there would soon be many photographs of a single pest, showing its appearance under many different situations. Analytical techniques can be used to cluster pest identification records into categories of similar features, and may even be able to identify trends and causes of pest outbreaks. The Java applet will evolve to include an expert system to help gather information which is important for identification, and may eventually include a complete diagnostics key. The taxonomic structure of the OODBMS facilitates categorizing pests, and photographs can be stored in attributes. For example, egg, larva, pupa and adult are the stages of development of an insect, and photographs of each stage can be stored in attributes of the same name. In addition, the integration of the photographs with pest management information (see earlier section) is easily facilitated.

Southern Trees CD-ROM

A database of thousands of plants, including descriptive data, text, photographs, and expert systems rules for selection of plants, has also been implemented using objects. Each object describes a plant, and contains many attributes such as soil requirements, growth habits, shape, flower color, pest problems, and other details. For example, Southern Trees CD-ROM (Beck, Gilman and Fowler, 1994) contains a database of 800 trees suitable for planting in urban areas of the southern United States. Each tree is described by over 50 different attributes. An expert system containing over 300 rules is used to help identify the characteristics of a particular site where the user wants to plant a tree. A query object is created based on these characteristics. The database of trees is searched to retrieve trees which match this query object. A list of these trees is returned to the user as a recommendation on what is best to plant at that site.

On-Line Thesaurus/Natural Language Project

The long term advantage of using OODBMS and semantic data models for building large knowledge bases will come in exploiting the semantics inherent in the structure of the database. Query processing which uses object matching techniques can do this now to some extent. Machine learning techniques such as conceptual clustering, induction of new classes, and case-based reasoning further utilize the content of objects to discover new knowledge.



- Select a term from from the pull-down menu (the menu contains nearly 600 terms related to lawn care).
- Press the "OK" button. The term will then appear on the display, with links to related terms. An example is shown in the display below.
- You can click on terms appearing in the graph to expand to neighboring terms.
- Click on a blue arrow ▶ by a term to go directly to an information resource.

Figure 2. On-line thesaurus Java applet. Each term is stored as an object in the Candide web server OODBMS.

If one thinks of an object as representing word meaning, then there is a basis for natural language processing which exploits the meaning of terminology inherent in the database. Although building a suitable dictionary would be a huge task (such a dictionary would contain tens of thousands of words and at least as many relationships among words), the OODBMS is a suitable platform for representing lexical knowledge and integrating it with other information in the database. Some exploratory work in using Candide to develop and store a lexicon of agricultural terms has already been done (Beck, 1991, Beck and Kumar, 1998).

Though we are a long way from a system which can understand and process queries stated in English, there are some short term practical results. One result is an on-line thesaurus of agricultural terms (Figure 2). This is a Java applet (http://hammock.ifas.ufl.edu/thesaurus) which can be used as a search tool to locate information about lawn care. This thesaurus contains 500 terms, and the terms are organized according to three abstractions: broader terms, narrower terms, and related terms. Users begin a search by selecting one of the 500 terms from a pull-down menu. A graph appears which shows the term and its neighboring broader, narrower, and related terms. Users can browse through this network by clicking on and expanding any of the terms. Arrows next to a term are links directly to information resources such as publications or photographs. Each term in the thesaurus and links to related terms are represented by an object in the database. The Java applet works by querying the ODBMS on a Web server each time the user requests more information.

The goal is to build an ontology of agricultural concepts that cover all the information from IFAS. This can be used as a search tool, but can also assist in development of a lexicon of agricultural terminology.

Conclusions

The recent evolution of object-oriented programming languages and databases on the Web is facilitating the deployment of these applications described above, many of which have been under development and use for many years. While the utility of the OODBMS approach has been widely demonstrated by these applications, there is a need for a large scale, robust OODBMS which is capable of managing a knowledge base for the entire organization. In addition to being large and diverse, IFAS is geographically distributed throughout every county in Florida. Currently, Candide is being implemented on top of a commercial OODBMS in order to meet these needs. Using CORBA and the Internet enables statewide distribution of database applications. Experience with commercial OODBMS products so far shows that they provide some excellent storage management, transaction management, backup and recovery facilities, as well as query processing abilities. However, they do not support abstract semantic data models. Rather the database schema is defined using Java or C++ class declarations. Thus it is desirable to implement Candide, or other suitable semantic data model, on top of these commercial products in order to provide a cleaner, highér level of data abstraction.

References

Beck, Howard W., Sunit K. Gala, and Shamkant B. Navathe. 1989. Classification as a query processing technique in the Candide semantic data model. Proceedings of the Fifth International Conference on Data Engineering. IEEE. Los Angeles, CA. pp. 572-581.

Beck, Howard W. 1991. Language acquisition from Cases. Proceedings of the DARPA Case-based Reasoning Workshop. Morgan Kaufmann, Inc. San Mateo, CA. pp 159-169.

Beck, H. W., E. F. Gilman, and P.A. Fowler. 1994. An Expert System for Tree Selection in Urban Forestry. Applied Engineering in Agriculture. 10(5):743-747.

Beck, Howard W., Tarek Anwar, and Shamkant Navathe. 1994. A conceptual clustering algorithm for database schema design. IEEE Transactions on Knowledge and Data Engineering . 6(3):396-411.

Beck, H.W. and B. Kumar. 1998. Clustering lexical patterns obtained from a text corpus. FLAIRS-98. Diane J. Cook. Ed. AAAI Press. Menlo Park, CA. pp. 304-308.

Williams, D.B., and H.W. Beck. 1996. SGML versus semantic data models in building digital agricultural libraries. Proceedings of the Ninth Florida Artificial Intelligence Research Symposium. Key West., FL. FLAIRS, St. Petersburg, FL. pp.405-409.

Database Design in Oracle With Srf As An Example*

Maxim Chekmasov, Marina Chekmasova

High Performance Database Research Center
School of Computer Science
Florida International University
University Park, Miami, FL 33199 U.S.A.
hpdrc@fiu.edu, http://hpdrc.cs.fiu.edu

1. INTRODUCTION

In recent years, Oracle relational database management system (RDBMS) has proved to be a powerful package for data storage and manipulation for a large variety of applications starting from PC-based desktops up to huge commercial transaction processing applications. Oracle continues to evolve, providing sophisticated storage, retrieval and distribution functions to enterprise-wide data processing and information management systems. There is a large number of Oracle related publications. We refer to [1-5], which cover topics on the database design.

The main goal of the present paper is to make an overview of the basic steps to be performed in the database design. As with all types of applications, the process of the database design begins with requirement analysis. This phase basically answers questions regarding what data elements must be stored, who will access them, and how. The second major step is to define the logical database. This phase is aimed to find out how information is grouped logically. The requirements are represented in the form of a model or schema, representing data in terms of business categories and relationships. On this stage a semantic approach to the database design, worked out at High Performance Database Research Center (HPDRC) proved to be an efficient tool for designing relational databases. Physical design is the next stage, in which individual data elements are given attributes and are defined as columns in tables. This phase also deals with performance considerations relating to the creation of indexes, rollback segments, temporary segments, and the physical layout of data files on disk.

We illustrate the database design process by a part of the joint project between HPDRC and the Everglades National Park (ENP) on a construction of the Everglades Environmental Database (EEDB). This part of the database stores the data on the Systematic Reconnaissance Flights (SRF). The flights are conducted within the ENP to study alligator nests, white tail deer and wading birds locations and activity.

2. DEFINITION OF REQUIREMENTS

System requirements are typically gathered through a series of interviews with the end users. This is an iterative process, which allows the database designers to find out the structure of the future database, basing on the document studies and feedback from the users. The process of constructing the logical schema of the database often overlaps with the requirement definition. The design is driven by the requirements. From the other side the logical model may bring out new requirements, that were not recognized in the earlier phases of the analysis. However, it is highly recommended to identify all the requirements before developing a physical design, since the cost of the underestimating the requirements after purchase of the hardware could be very high.

A common way used to define and document database requirements is to develop a data dictionary, which defines the data elements to be stored. Data dictionary has obvious drawbacks. It does not describe how the individual items are related. It also lacks information regarding how the data is created, updated, and retrieved, among other things. Below we present some elements of the data dictionary, describing the SRF.

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

Item	Description
Starting date	The date when the survey was started
Survey type	Same methodology is applied to b – birds, d – deer, a – alligator, o – other species and substances observed
Survey height	The height in which the aircraft was flying from the ground at the time of observation. Measured in feet.
Survey strip width	Width of the area observed. Measured in meters.
Ground speed	Speed with which aircraft was flying at the time of observation. Measured in miles per hour.
Comments	Comments on the SRF survey
. . . . .	. . . . .

A functional specification describes the system requirements in plain English, and explains in details who will be using the system, when and how. Information concerning the number of concurrent users accessing the system, how frequently records are inserted and updated, and how information will be retrieved are the topics to be covered in the functional specification. These factors will help to determine hardware and software licensing requirements, and will have an impact on performance, security, and database integrity issues. Here is a possible example of the functional description for the SRF:

The database will be available to 1 system administrator, 1 Oracle database administrator, a group of 5 persons, responsible for collecting data, 2 technicians and about 30 other users, for a total of 39 users. Of these 39 users, it is expected that a maximum of 6 would be actively using the system at any given time. Only 9 users may add information regarding new SRF surveys. Only 3 users are able to update the already loaded records. . . . . .

In the example we've got some data elements regarding access and security. As a rule the functional specification and data dictionary are developed simultaneously, as one document may provide relevant information that should be reflected in the other.

It should be taking into account, that the users will not be able to fully explain the system requirements on their own. The database designers should do their best to describe the system to the fullest extent and detail possible. Poor requirement definition will most likely result in poor or inadequate design. This phase of the development process should not be underestimated.

3. THE LOGICAL MODEL

One of the ways to represent the logical model is through a Category-Relationship (C-R) diagram. A category is defined as a discrete object for which items of data are being stored, and a relationship refers to an association between two categories. In the SRF example, we have the following categories and the relationships between them:

Category	Relationship
Organization (responsible for conducting the survey)	One, or many SRF may have one responsible organization
Aircraft type (type of the aircraft, used to perform the flights)	One, or many surveys may use an aircraft of the particular type
Project (within which the survey was conducted)	One, or many surveys are conducted within the project
SRF Survey (general information about the survey)	
Flight (conducted within the survey)	One flight is conducted within one particular SRF survey
Personnel (the crew, conducted the flight)	One, or many flights may have one employee as right observer during the flight. One, or many flights may have one employee as left observer during the flight. One, or many flights may have one employee as a pilot during the flight.
Subflight (a part of the flight along the transect)	One subflight is a part of one flight
Transect (information concerning transects)	One subflight is carried along one transect
Survey cell (information concerning cells)	One survey cell is determined by one transect

Now we are ready to make a step further by defining the attributes for the categories. The attributes are the individual items of data to be stored that relate specifically to the object. The resulted model is good at representing basic data concepts but is not of much use, when it comes to physical implementation. We should employ the relational model to bridge this gap.

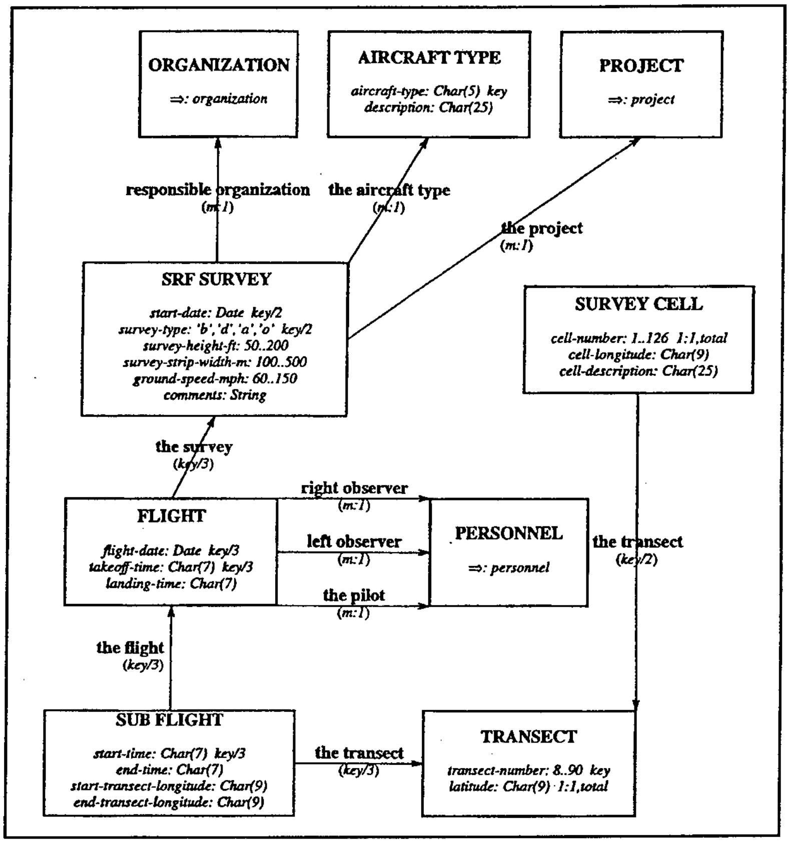
The relational model is characterized by use of keys and relations, among other things. In the context of relational database theory a relation can be viewed as an unordered, two-dimensional table, where each row is distinct. Relationships are built between relations (tables) through common attributes. These common attributes are called keys. Among the keys a primary key (PK) uniquely identifies a row in a relation and each relation may have only one primary key. In the case when more than one attribute uniquely identify each row in a relation, the aggregate of these attributes is called a composite key. A foreign key (FK) exists only on terms of the relationship between two relations. A foreign key in a relation is a nonkey attribute that is a primary key (or part of the primary key) in another relation. This is a shared attribute that forms a relationship between two relations (tables).
Below, we present the attributes for the category SRF Survey before and after employing relational model:

Before employing relational model	**After employing relational model**
Start date	Start date (PK)
Survey type	Survey type (PK)
Survey height	Survey height
Survey strip width	Survey strip width
Ground speed	Ground speed
Comments	Comments
	Responsible organization (FK) Aircraft type (FK) Project (FK)

4. SEMANTIC APPROACH TO THE DESIGN OF THE LOGICAL MODEL

Semantic approach to the database design worked out at HPDRC can be effectively implemented for the design of the relational databases. One of the key advantages of the semantic approach is that it highlights the functional logic of the constructed database. Another strong side of this approach is in its clear and straightforward appearance. It proves to be understandable by a person, who is not familiar with the theory of databases and storing data in the computer. This person usually represents a typical user of the future database.

Let us turn to the SRF example. The schema below represents a semantic design of the SRF. The categories are drawn in the boxes. The name of a category is written in capital letters and the name of its attributes and ranges are written in lowercase italic. The arrows represent the relations between the categories. Each relation has name.



Briefly this schema can be explained as follows. The SRF-SURVEY is identified by the start-date and the survey-type. The survey is performed within the PROJECT by the responsible ORGANIZATION. The survey consists of a set of FLIGHTs, performed on the aircraft of some particular AIRCRAFT-TYPE. The pilot and two observers, who are described within the category PERSONNEL, represent the flight crew. The flight, in turn, is a set of SUBFLIGHTs. Each SUBFLIGHT is performed along some particular TRANSECT, which has its unique number and is determined by latitude. The transect determines a set of SURVEY-CELLs. The location of each particular cell is uniquely identified by the cell-longitude and the latitude of the corresponding transect. Thus, the cells determine a grid, which covers the territory of the Park.

Construction of the semantic schema is a mean for visualizing a logical model of the database. This graphical presentation allows to eliminate logical errors in the design and also serves as a part of documentation describing the database. We refer to [6] for the detailed discussion of the semantic modeling approach.

The design of the tablespaces can be considered as an aspect of the logical model. A TABLESPACE consists of one or more data files and houses one or more database objects. Before proceeding to the physical design, designers should consider how they might want to use tablespaces to group database objects along logical boundaries. Below we create the tablespace for SRF. Later we'll probably need to use ALTER TABLESPACE command to add data files to the tablespace.

```
CREATE TABLESPACE wildlife DATAFILE '/disk01/oradata/wild1.dbf' SIZE 20M;
ALTER TABLESPACE wildlife ADD DATAFILE '/disk01/oradata/wild2.dbf' SIZE 10M;
```

5. THE PHYSICAL MODEL

The physical database consists of data files, tablespaces, rollback segments, tables, columns, and indexes. There are dependencies between these elements that impose an order on the design process. The process usually starts with designing the smallest units of physical storage (the column) and proceeds to larger unit of storage. As with logical modeling, developing a physical design is an iterative process.

The design of the physical database begins with assigning column attributes. The data type and length of a column should be carefully chosen at design time, because it is sometimes difficult to change these attributes after data has been loaded. Besides the nature of the data and its length, additional factors should also be considered while choosing a data type for a column. For example, two VARCHAR2 values must be the same length to be considered equal, where two CHAR values are compared without consideration of trailing spaces. Also defining the column attributes is an important step in capacity planning. From this information, the maximum record size for each table can be determined. This combined with an estimate of the total number of rows helps determine the amount of storage required to house the data.

The next step is to begin writing Data Definition Language (DDL) scripts that will be used to create the tables. The DDL for creating tables consists of defining column attributes and constraints, storage specification and table constraints. We'll discuss constraints later. For now, let us concentrate on column attributes and storage specification. We illustrate these steps with the **srf_survey** table creation script:

```
CREATE TABLE srf_survey (
     start_date_K            DATE
   , survey_type_K           CHAR(1)
   , survey_height_ft        NUMBER(*,0)
   , survey_strip_width_m    NUMBER(*,0)
   , ground_speed_mph        NUMBER(*,0)
   , comments                VARCHAR(250)
   , r_organization__name    VARCHAR2(50)
   , aircraft_type           VARCHAR2(5)
   , project_id              VARCHAR2(30)
)
TABLESPACE wildlife
STORAGE ( INITIAL          1M
          NEXT           100K
          MINEXTENTS        1
          MAXEXTENTS       10
          PCTINCREASE      10 );
```

The STORAGE specification of the DDL indicates that 1 megabyte will be allocated initially for storing data in the **srf_survey** table. The additional extents will start at 100 kilobytes. There will be a minimum of 1 extent and a maximum of 10 extents. Each extent will be 10 percent larger than the previous extent. This specification will allow the table to grow to about 2 megabytes. This information is vital for capacity planning. Although storage specifications can be modified using ALTER TABLE, it is recommended to allow for the maximum estimated size or more. It is better to overestimate storage requirements than to underestimate them.

When the TABLESPACE clause is omitted from CREATE TABLE statement, the table is created in the default tablespace. Analogously, if the STORAGE parameter is omitted the table is created with the default storage parameters.

6. INTEGRITY OF THE DATABASE

Oracle provides many ways to enforce integrity, including column constraints, table constraints, sequences, and triggers. Column constraints are widely used means of enforcing integrity. The NOT NULL column constraint ensures that null values are not inserted into a column. The NOT NULL constraint can be used in conjunction with the UNIQUE constraint. The UNIQUE constraint usually designates a secondary key and causes Oracle to automatically create an index. The UNIQUE constraint does not prevent null values from being inserted. For this reason it is often used for columns that should be unique but do not necessarily need to be populated.

The most flexible column constraint is a CHECK constraint. The CHECK constraint can reference any column in the table. It cannot reference any external objects, system variables, or system constants. When the CHECK constraint references columns, the conditions are always applied to the current row. In order to insert or update a column with the CHECK constraint, the specified condition must evaluate to TRUE or unknown (when a NULL value is being inserted into one of the columns referenced by the CHECK condition). CHECK can be used in conjunction with other column constraints. For the table srf_survey we have the following column constraints:

```
ALTER TABLE srf_survey  ADD CONSTRAINT survey_type
                        CHECK ( survey_type_K        IN ('b','d','a','o'));
ALTER TABLE srf_survey  ADD CONSTRAINT survey_height
                        CHECK ( survey_height_ft      BETWEEN  50 AND 200 );
ALTER TABLE srf_survey  ADD CONSTRAINT strip_width
                        CHECK ( survey_strip_width_m BETWEEN 100 AND 500 );
ALTER TABLE srf_survey ADD CONSTRAINT ground_speed
                        CHECK ( ground_speed_mph     BETWEEN  60 AND 150 );
```

PRIMARY KEY is the most significant among the constraints. It is used to ensure that each row in the table is unique. When a column is declared as PRIMARY KEY, the additional constraints UNIQUE and NOT NULL are implied. Also an index on the column(s) is automatically created and assigned a unique name by Oracle. Each of the column constraints described above can also be applied as table constraints, with the exception of NOT NULL. Table constraints have the additional advantage of being able to operate on multiple columns. For the **srf_survey** table, for example, the PRIMARY KEY constraint affects two columns and appears to be a table constraint:

```
ALTER TABLE srf_survey ADD PRIMARY KEY (start_date_K, survey_type_K);
```

Let us turn to the discussion of referential integrity issues. Referential integrity is a condition in which all references to external objects within each database object are valid. Enforcing referential integrity is a critical task in ensuring that data is accurate and complete. Referential integrity problems can result in data loss, wasted storage, and inaccurate data.

Using the REFERENCES keyword column constraints can be employed to enforce referential integrity for foreign keys. The table and column specified in the REFERENCES clause must already exist, and the referenced column must be defined as a UNIQUE or PRIMARY KEY. The REFERENCES column constraint does not imply NOT NULL, but NOT NULL can be used in conjunction with a REFERENCES constraint on a column. Also the data type is unnecessary for columns containing a REFERENCES constraint. The column will automatically be defined with the data type of the column it references. In our example with **srf_survey**, this table is related to **organization**, **aircraft_type** and **project** tables by the corresponding columns. To ensure the integrity, we perform the following commands:

```
ALTER TABLE srf_survey ADD CONSTRAINT organization_fk
      FOREIGN KEY (r_organization__name) REFERENCES organization (name_key);
ALTER TABLE srf_survey ADD CONSTRAINT aircraft_fk
      FOREIGN KEY (aircraft_type)        REFERENCES aircraft_type (aircraft_type_key);
ALTER TABLE srf_survey ADD CONSTRAINT project_fk
      FOREIGN KEY (project_id)           REFERENCES project (project_id_key);
```

7. PERFORMANCE CONSIDERATIONS

When designing the physical database, performance is an important consideration. There are numerous factors related to the design that will affect the overall performance of the database. These factors include the data model itself, indexing, rollback and temporary segments, and the physical location of the data on the disks.

Let us turn to the indexes first. Indexes can be created on single or multiple columns, and may or may not be unique. When creating an index on multiple columns, the order in which the columns are declared is particularly important, because Oracle treats the values of such an index as an aggregate. The column that will be used the most should be declared first in a multicolumn index. The creation of indexes should be planned very carefully, because improper use of indexes can have a damaging effect on performance. Even where the indexes improve the performance of SELECT statements, they have a negative impact on INSERTs and UPDATEs, because the indexes must be modified in addition to the tables.

In our example, suppose we decided to create a separate tablespace **wildindx** to store the indexes of our database. Then in the case of the **srf_survey** table we should drop the already created PRIMARY KEY constraint **survey_pk** and re-create it with the instruction to store index in the tablespace **wildindx**. Here are the commands:

```
CREATE TABLESPACE wildindx    DATAFILE '/disk02/oradata/phys1.dbf' SIZE 20M;
ALTER TABLE            srf_survey   DROP CONSTRAINT survey_pk;
ALTER TABLE            srf_survey
     ADD CONSTRAINT survey_pk
          PRIMARY KEY (start_date_K, survey_type_K)
          USING INDEX
          TABLESPACE wildindx
          STORAGE  (INITIAL 1M NEXT 100K MAXEXTENTS 100 PCTINCREASE 10);
```

The column attributes play a role in performance as well. Wherever possible, integers should be used as keys because they can be compared faster than any other data type. From the point of improving performance, column and table constraints should be avoided, if possible, because they must be checked whenever the value is inserted or updated.

Rollback segments also play an important role in the overall performance of the database. Oracle uses rollback segments as temporary storage for data needed to reverse a transaction. This data must be stored until the transaction is committed. Rollback segments must be sufficiently large to store this data for all transactions occurring at a given time. If rollback segments are not large enough, transaction will fail. To properly estimate the size of the rollback segments needed, the designer must know how many users will be submitting transactions, and the maximum size of the rows affected by a single transaction. A rollback segment, like other database objects, can be created with a script. For the SRF we create a separate tablespace for rollback segments first, then we create a segment and bring it ONLINE:

```
CREATE TABLESPACE rbs DATAFILE '/disk01/oradata/rbs.dbf' size 80M;
CREATE ROLLBACK SEGMENT r01
          TABLESPACE rbs
          STORAGE ( INITIAL 1M     NEXT 1M     MINEXTENTS 2     MAXEXTENTS 79);
ALTER ROLLBACK SEGMENT r01 ONLINE;
```

Another performance consideration relates to the creation of temporary segments. Temporary segments are similar to rollback segments, except that they are used to store result sets rather than transaction information. When a SELECT statement produces a result set that is too large to be stored in memory, a temporary table is created to store the results until the cursor is closed. Temporary tables may also be created by Oracle to store temporary result sets for complex joins or unions. As with rollback segments, these temporary segments must be sufficiently large to store this data, or select statements may fail. Other performance considerations relate to the physical layout of files on disks. Proper use of multiple disks and controllers, clustering, and striping can improve performance greatly in certain situations.

8. SOME DATABASE SECURITY ISSUES

Suppose we've created Oracle account **srf** and **srf_survey** table under this account. Also let us assume, that tables **organization** and **project** already exist under **common** account, which contains the tables, storing general information, such as a list of organizations and a list of projects, conducted within the Park. Then the **srf_survey** table is related to the tables in another Oracle account. To preserve the referential integrity of the whole database, we should perform the following commands:

```
In common account:
GRANT SELECT, REFERENCES ON organization TO srf;
GRANT SELECT, REFERENCES ON project TO PUBLIC;
In srf account:
CREATE SYNONYM organization FOR common.organization;
CREATE SYNONYM project FOR common.project;
```

In this example SELECT and REFERENCES privileges are granted to **srf** account on **organization** table, and these privileges are granted to all accounts on **project** table. In **srf** account the corresponding synonyms are created. The granted privileges allow performing only SELECT statements on **organization** and **project** tables under **srf** user account, no INSERTs and UPDATEs are allowed.

9. CAPACITY PLANNING

Knowing the size of column attributes, determine size of each row in the table. The column attributes also determine the size of each row in indexes created on the columns. The attributes, combined with the estimated total number of rows (including provisions for future growth), are used in defining the storage clause for tables and indexes. For purpose of capacity planning, it should be assumed that all objects would reach their maximum extents. The next step is creating DDL for tablespaces. The data file(s) created by these scripts should be sufficiently large to contain all objects that they will contain. The total size of the database can then be determined by simply adding the sizes of the data files.

In capacity planning, the designer must accommodate for unexpected growth. Usually at least 25 percent (preferably 50 percent) of each disk should be free after the initial installation. This will allow additional data files to be created wherever necessary if tables grew larger than expected. The importance of capacity planning should not be underestimated.

SELECTED BIBLIOGRAPHY

1. G.Koch, K.Loney *Oracle8: The Complete Reference (Oracle Series)*, Oracle Press, 1997, 1300 p.
2. K.Loney *Oracle 8 Dba Handbook*, Osborne McGraw-Hill, 1997, 822 p.
3. M.J.Corey (ed.) *Oracle8 Tuning*, Osborne McGraw-Hill, 1997, 293 p.
4. *ORACLE Magazine*, ISSN 1065-3171, Oracle Corporation, http://www.oramag.com; http://www.oracle.com.
5. R.Becker, et.al. *Oracle Unleashed*, SAMS Publishing, 1996, 1404 p.
6. N.Rishe *Database Design: The Semantic Modeling Approach*, McGraw-Hill, 1992, 528 p.

An Introduction to Designing Database Relationships with ORN

Bryon K. Ehlmann and Angela I. Bowers

CIS Department, Florida A&M University, Tallahassee FL 32307, USA
{ehlmann, abowers}@cis.famu.edu

Abstract. **Object Relationship Notation (ORN) is a declarative scheme for defining common relationship semantics. The ORN Simulator is a tool that allows one to become adept at relationship design and to easily model relationships with ORN. Users can define relationships and then observe and fine-tune their behavior as they create and delete objects and create, destroy, and change relationship instances within a prototype database. Once relationship behavior has been defined and verified using the ORN Simulator, a user can easily implement this behavior in a real database application by using a Database Management System (DBMS) that supports ORN. The implementation requires no programming and no specification of complex constraints and triggers. In this paper, we give a brief introduction to ORN by showing how it is used in the ORN Simulator to design database relationships. We also show how it is incorporated into an Object DBMS and provide references to other papers that cover ORN in more detail.**

1. Introduction

Object Relationship Notation (ORN) provides a declarative scheme for defining a large variety of common *aggregate* relationship types—i.e., the "is associated with," "is defined by," "is owned by," and "is a part of" types of relationships and their many variations [1, 4]. Such types define the boundaries for *complex* and *composite* objects in a database [10]. ORN allows the semantics, or behavior, of these relationship types to be identified and documented during system analysis and design and to be defined to a DBMS during implementation. This facilitates the early detection of relationship subtleties and inconsistencies and the automatic maintenance of proper relationship behavior by the DBMS. Significantly, this is achieved without having to develop any programming code or complex constraint and trigger specifications [8].

Previous papers have explored various aspects of ORN. In [4] an early version of ORN was presented and used to model relationships in a scientific database. In [6] ORN was compared to other declarative schemes for specifying relationship semantics, e.g., the REFERENCES clause for foreign keys in SQL. [6] showed that the most unique (and powerful) feature of ORN was that it provides for the enforcement of upper and lower bound cardinality constraints and allows delete propagation to be based on these constraints. [5] presented a integrated methodology for developing relationships in a database based on ORN. [7] discussed its implementation in an extensible Object ODBMS (ODBMS) prototype called Object Relater *Plus* (OR+). [9] showed that with ORN, subtleties and inconsistencies in relationship behavior can be identified and automatically detected during analysis and design. In [8] the syntax, semantics, and pragmatics for incorporating ORN into SQL were described as well as the benefits. And finally, in [2] the user interface, architecture, and features of the ORN Simulator were first presented.

In the remainder of this paper, we give a brief introduction to ORN by showing how it is used in the ORN Simulator and conclude by showing how it can be incorporated into an ODBMS.

2. ORN Simulator

The ORN Simulator is a practical tool to assist database designers and help students of database design become more familiar with relationship design and ORN. The tool allows a user to model the relationships in their database application. Relationship types between object classes are defined using ORN. Then, relationship behavior is observed as objects are created and deleted and relationships between

Then, relationship behavior is observed as objects are created and deleted and relationships between these objects are created, destroyed, and changed. If necessary, the user fine-tunes the behavior by redefining the relationship type, again via ORN. The user performs these operations on a small prototype database by simply pointing and clicking on menu selections, classes, relationships, objects, and relationship instances in an ER-like Diagram [3]. The ORN Simulator Help screen for "Using the Program..." is given in Fig. 1 and explains these operations in detail.

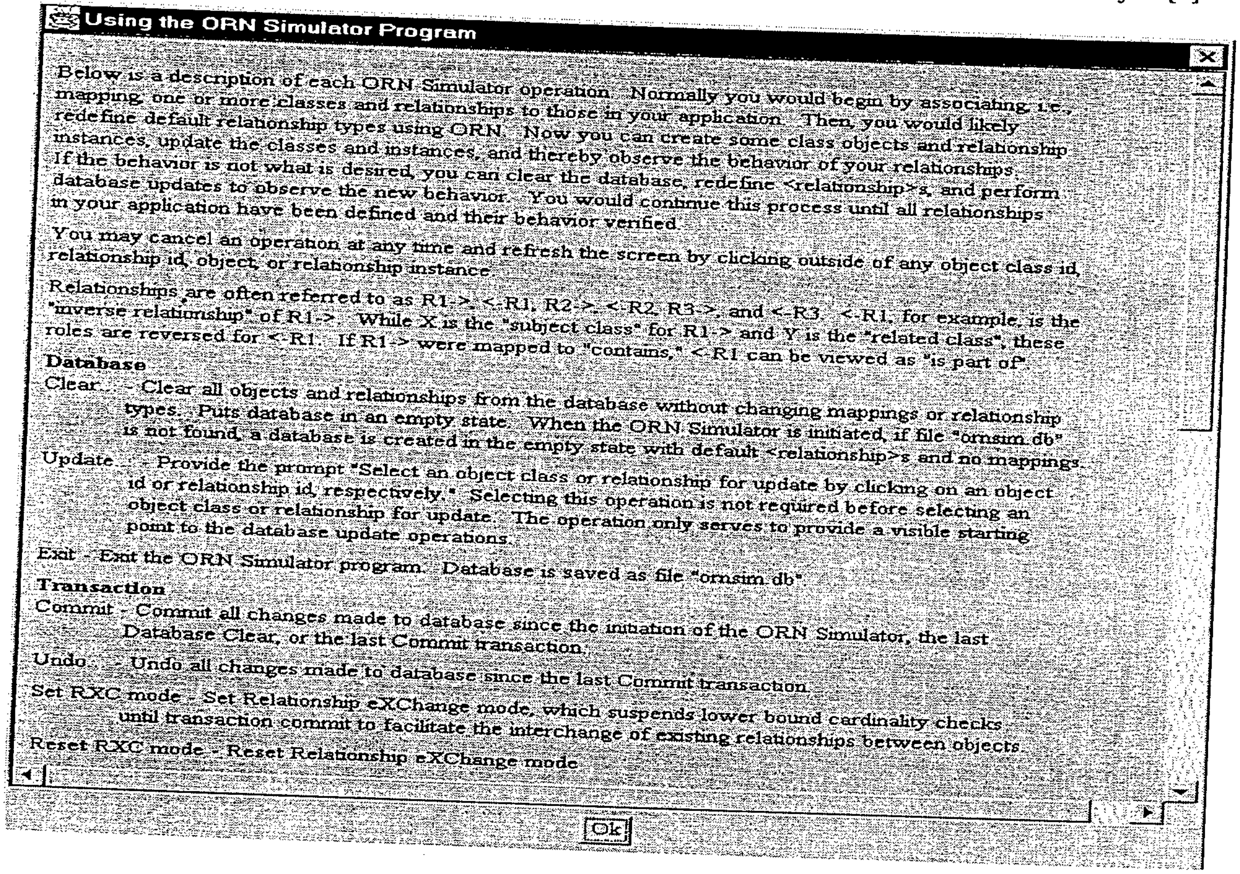
The Help screen for "ORN_Syntax and Semantics..." is given in Fig. 2. ORN defines relationships in terms of the *cardinalities* and *bindings* for the *subject* and *related* classes. For example, the relationship between employees and car pools can be defined as |~X~<2..-to-0/1>. The binding and cardinality given first apply to the subject class, those given second apply to the related class. In the given *<relationship>*, binding |~X~ and cardinality 2.. apply to the Employee class, and default binding and cardinality 0/1 apply to the Car Pool class. Each car pool is related to 2 or more employees (lower bound 2, upper bound infinity), and each employee is related to 0 or 1 car pools (lower bound 0, upper bound 1). After a *<relationship>* is "Redefined" in the ORN Simulator, its cardinalities and bindings are properly placed onto the ER Diagram (see Fig. 3.).

Bindings indicate the level of binding between related objects. The level of binding determines the implicit and explicit destructibility of relationship instances and if relationship destruction can result in the implicit deletion of related objects. In the |~X~<2..-to-0/1> relationship, the |~X~ binding denotes that implicit, denoted by |, and explicit, denoted by X, destruction of a relationship instance can result in the implicit deletion of a car pool. Implicit destructibility of relationships is relevant to object deletion. All existing relationships involving an object must be implicitly destroyed, or cut, before an object can be deleted. Implicit deletions of related objects resulting from relationship destruction enforce cardinalities and define the extent of complex and composite objects.

Fig. 3 shows the results of deleting an employee in a simulated database application. The top screen shows the database after the user has already mapped object classes and relationships to those in a company personnel database, defined desired *<relationship>*s, and created some objects and relationship instances. The bottom screen shows the database after the user has clicked on class Y, selected "Delete an Object..." from a popup menu, and clicked on object y2 in the Y rectangle. y2 has been deleted. Note also that instance x2<-->y2 has been implicitly destroyed. This behavior is prescribed by the default binding and 0/M cardinality for the Employee class in the <1-to-0/M> R1, or "has," relationship. Note also that instances y2<-->z0 and y6<-->z0 have been implicitly destroyed and car pool z0 has been implicitly deleted. This behavior is prescribed by the |~ binding and 2.. cardinality for the Employee class in the |~X~<2..-to-0/1> "belongs to" relationship. It enforces the semantic that a car pool "is defined by" two or more riders! If either y2<-->z0 or y6<-->z0 would have been explicitly destroyed, z0 would have also been implicitly deleted. This is prescribed by the X~ binding and 2.. cardinality.

By performing additional operations on objects and relationships, the ORN Simulator user can observe other types of relationship behavior based on the defined relationships. If employee y9 is created and no instance is created relating it to some X (Organization) object, then a cardinality exception occurs on the next transaction Commit. An employee must be assigned to an organization! If a delete is attempted on object x0, an exception occurs (resulting from the default binding and the 1 cardinality for Organization.) An organization cannot be removed if it has any employees! They must first be reassigned. If instance x0<-->x1 is destroyed, objects x1, x3, x8, and x9 are implicitly deleted. An organization is a composite object! x4, however, is not implicity deleted since it is related to x2. Separating (and thus eliminating) an organization from this company means eliminating all its subordinate organizations—that is, unless control of one of these organizations is shared by a remaining organization.

Messages that flag relationship inconsistencies may appear on the screen when the user Redefines a relationship—e.g., defining a <*relationship*> that on delete of a subject object requires an implicit delete of a related object that can never be deleted because of a defined <*relationship*> it has to another object [9].



Below is a description of each ORN Simulator operation. Normally you would begin by associating, i.e., mapping, one or more classes and relationships to those in your application. Then, you would likely redefine default relationship types using ORN. Now you can create some class objects and relationship instances, update the classes and instances, and thereby observe the behavior of your relationships. If the behavior is not what is desired, you can clear the database, redefine <relationship>s, and perform database updates to observe the new behavior. You would continue this process until all relationships in your application have been defined and their behavior verified.

You may cancel an operation at any time and refresh the screen by clicking outside of any object class id, relationship id, object, or relationship instance.

Relationships are often referred to as R1->, <-R1, R2->, <-R2, R3->, and <-R3. <-R1, for example, is the "inverse relationship" of R1->. While X is the "subject class" for R1-> and Y is the "related class", these roles are reversed for <-R1. If R1-> were mapped to "contains," <-R1 can be viewed as "is part of".

Database

Clear - Clear all objects and relationships from the database without changing mappings or relationship types. Puts database in an empty state. When the ORN Simulator is initiated, if file "ornsim.db" is not found, a database is created in the empty state with default <relationship>s and no mappings.

Update - Provide the prompt "Select an object class or relationship for update by clicking on an object id or relationship id, respectively." Selecting this operation is not required before selecting an object class or relationship for update. The operation only serves to provide a visible starting point to the database update operations.

Exit - Exit the ORN Simulator program. Database is saved as file "ornsim.db".

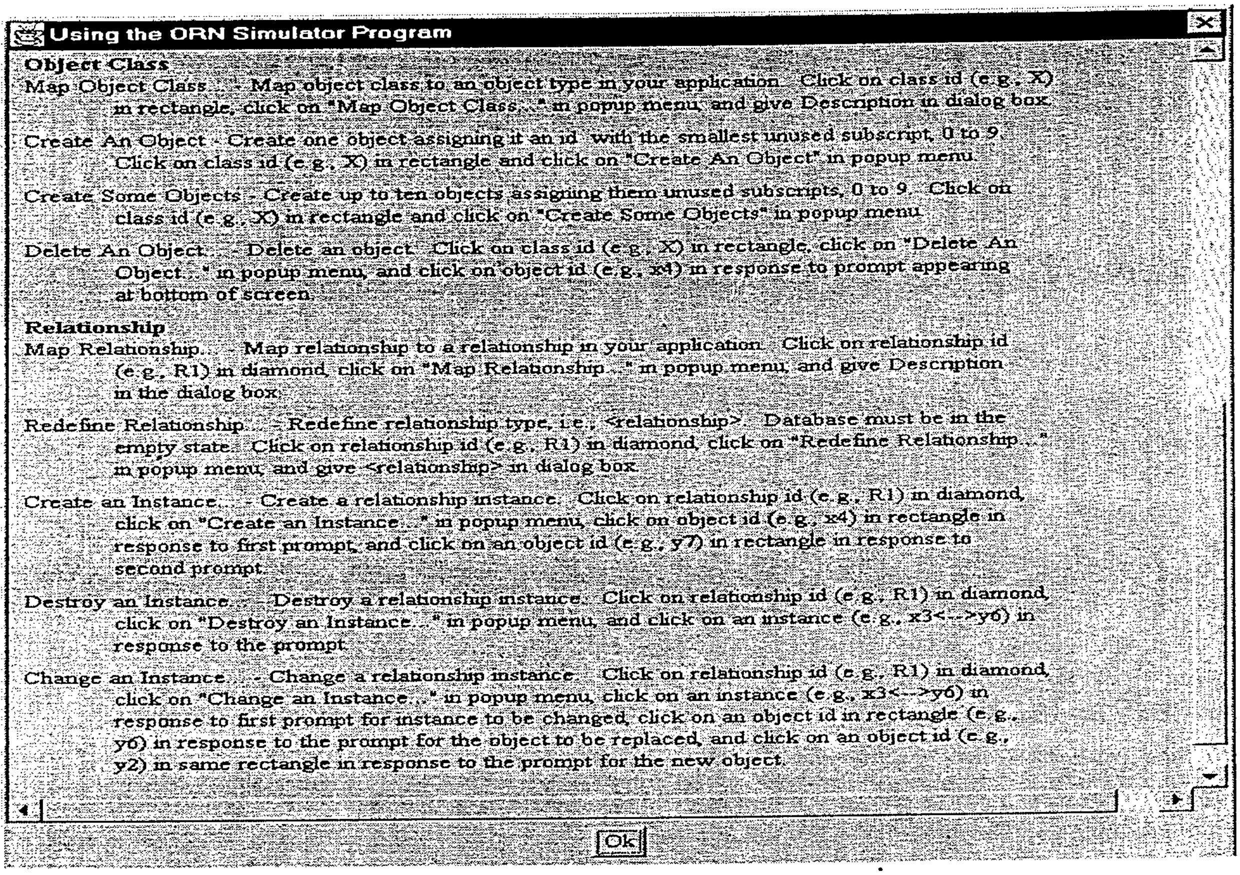
Transaction

Commit - Commit all changes made to database since the initiation of the ORN Simulator, the last Database Clear, or the last Commit transaction.

Undo - Undo all changes made to database since the last Commit transaction.

Set RXC mode - Set Relationship eXChange mode, which suspends lower bound cardinality checks until transaction commit to facilitate the interchange of existing relationships between objects.

Reset RXC mode - Reset Relationship eXChange mode



Using the ORN Simulator Program

Object Class

Map Object Class ... - Map object class to an object type in your application. Click on class id (e.g., X) in rectangle, click on "Map Object Class..." in popup menu, and give Description in dialog box.

Create An Object - Create one object assigning it an id with the smallest unused subscript, 0 to 9. Click on class id (e.g., X) in rectangle and click on "Create An Object" in popup menu.

Create Some Objects - Create up to ten objects assigning them unused subscripts, 0 to 9. Click on class id (e.g., X) in rectangle and click on "Create Some Objects" in popup menu.

Delete An Object... - Delete an object. Click on class id (e.g., X) in rectangle, click on "Delete An Object..." in popup menu, and click on object id (e.g., x4) in response to prompt appearing at bottom of screen.

Relationship

Map Relationship... - Map relationship to a relationship in your application. Click on relationship id (e.g., R1) in diamond, click on "Map Relationship..." in popup menu, and give Description in the dialog box.

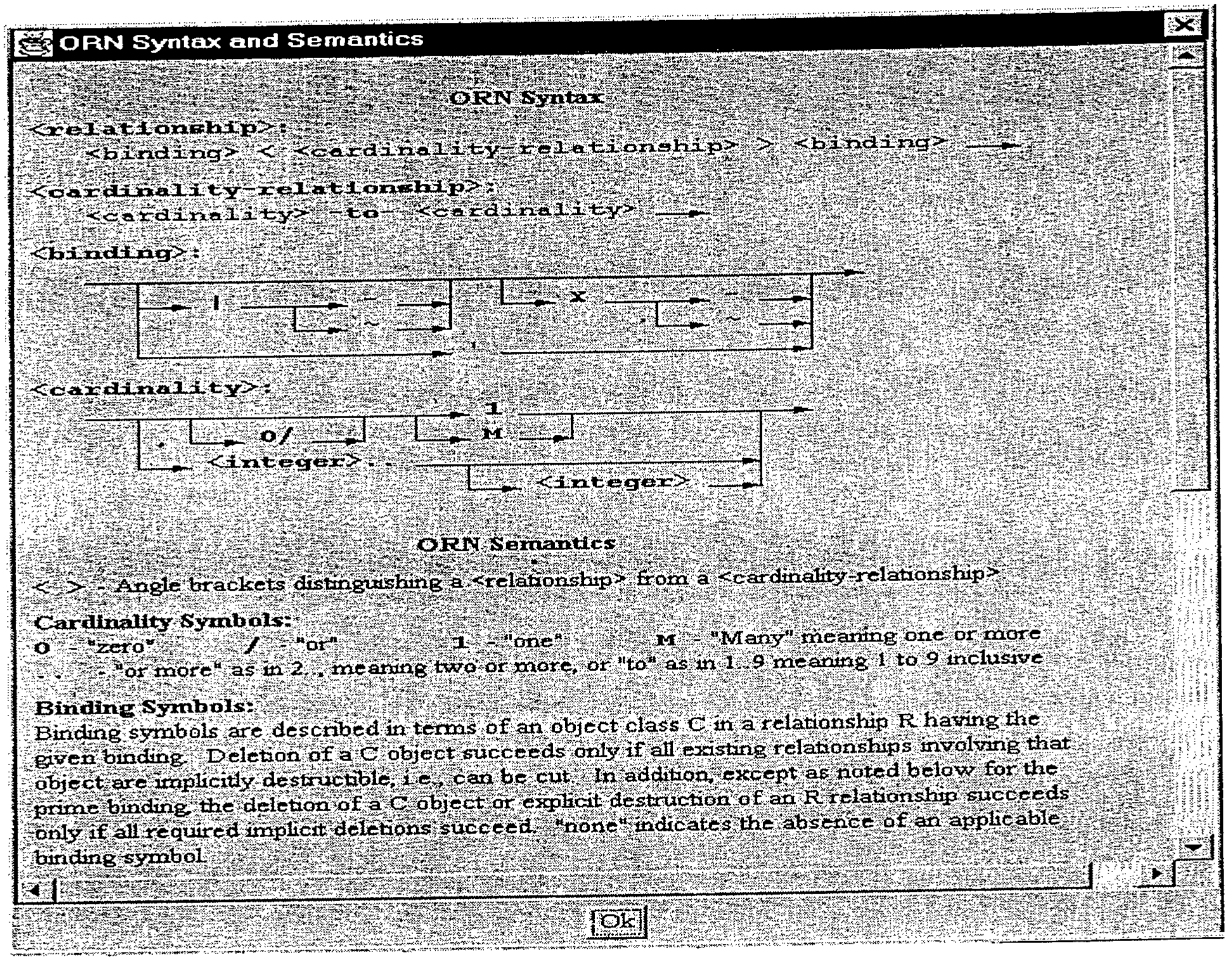
Redefine Relationship... - Redefine relationship type, i.e., <relationship>. Database must be in the empty state. Click on relationship id (e.g., R1) in diamond, click on "Redefine Relationship..." in popup menu, and give <relationship> in dialog box.

Create an Instance... - Create a relationship instance. Click on relationship id (e.g., R1) in diamond, click on "Create an Instance..." in popup menu, click on object id (e.g., x4) in rectangle in response to first prompt, and click on an object id (e.g., y7) in rectangle in response to second prompt.

Destroy an Instance... - Destroy a relationship instance. Click on relationship id (e.g., R1) in diamond, click on "Destroy an Instance..." in popup menu, and click on an instance (e.g., x3<-->y6) in response to the prompt.

Change an Instance... - Change a relationship instance. Click on relationship id (e.g., R1) in diamond, click on "Change an Instance..." in popup menu, click on an instance (e.g., x3<-->y6) in response to first prompt for instance to be changed, click on an object id in rectangle (e.g., y6) in response to the prompt for the object to be replaced, and click on an object id (e.g., y2) in same rectangle in response to the prompt for the new object.

Fig. 1. Using the ORN Simulator Program—Getting Started, Operations, and User Actions.



ORN Syntax

<relationship>:
<binding> < <cardinality-relationship> > <binding>

<cardinality-relationship>:
<cardinality> -to- <cardinality>

<binding>:

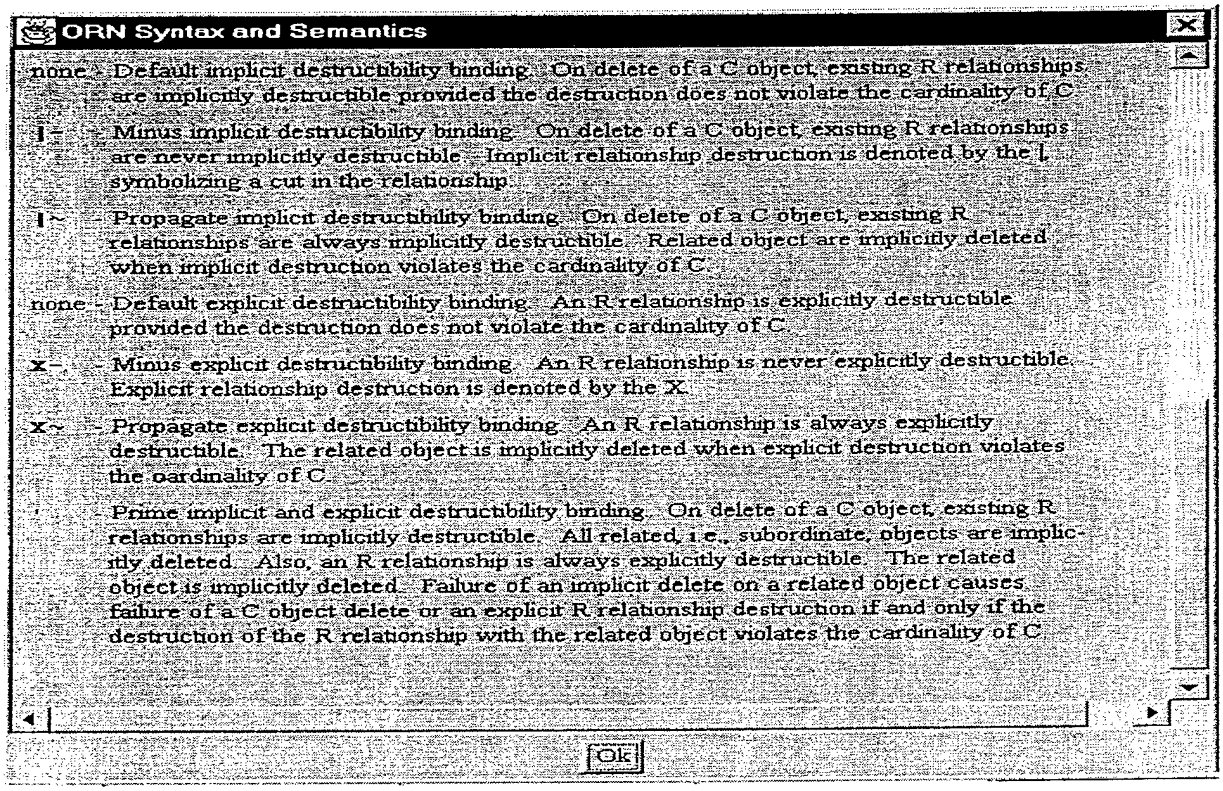
<cardinality>:

ORN Semantics

< > - Angle brackets distinguishing a <relationship> from a <cardinality-relationship>

Cardinality Symbols:
0 - "zero" **/** - "or" **1** - "one" **M** - "Many" meaning one or more
. . - "or more" as in 2.., meaning two or more, or "to" as in 1..9 meaning 1 to 9 inclusive

Binding Symbols:
Binding symbols are described in terms of an object class C in a relationship R having the given binding. Deletion of a C object succeeds only if all existing relationships involving that object are implicitly destructible, i.e., can be cut. In addition, except as noted below for the prime binding, the deletion of a C object or explicit destruction of an R relationship succeeds only if all required implicit deletions succeed. "none" indicates the absence of an applicable binding symbol.



none - Default implicit destructibility binding. On delete of a C object, existing R relationships are implicitly destructible provided the destruction does not violate the cardinality of C.

I - - Minus implicit destructibility binding. On delete of a C object, existing R relationships are never implicitly destructible. Implicit relationship destruction is denoted by the I, symbolizing a cut in the relationship.

I~ - Propagate implicit destructibility binding. On delete of a C object, existing R relationships are always implicitly destructible. Related object are implicitly deleted when implicit destruction violates the cardinality of C.

none - Default explicit destructibility binding. An R relationship is explicitly destructible provided the destruction does not violate the cardinality of C.

x- - Minus explicit destructibility binding. An R relationship is never explicitly destructible. Explicit relationship destruction is denoted by the X.

x~ - Propagate explicit destructibility binding. An R relationship is always explicitly destructible. The related object is implicitly deleted when explicit destruction violates the oardinality of C.

' - Prime implicit and explicit destructibility binding. On delete of a C object, existing R relationships are implicitly destructible. All related, i.e., subordinate, objects are implicitly deleted. Also, an R relationship is always explicitly destructible. The related object is implicitly deleted. Failure of an implicit delete on a related object causes failure of a C object delete or an explicit R relationship destruction if and only if the destruction of the R relationship with the related object violates the cardinality of C.

Fig. 2. ORN Syntax and Semantics.

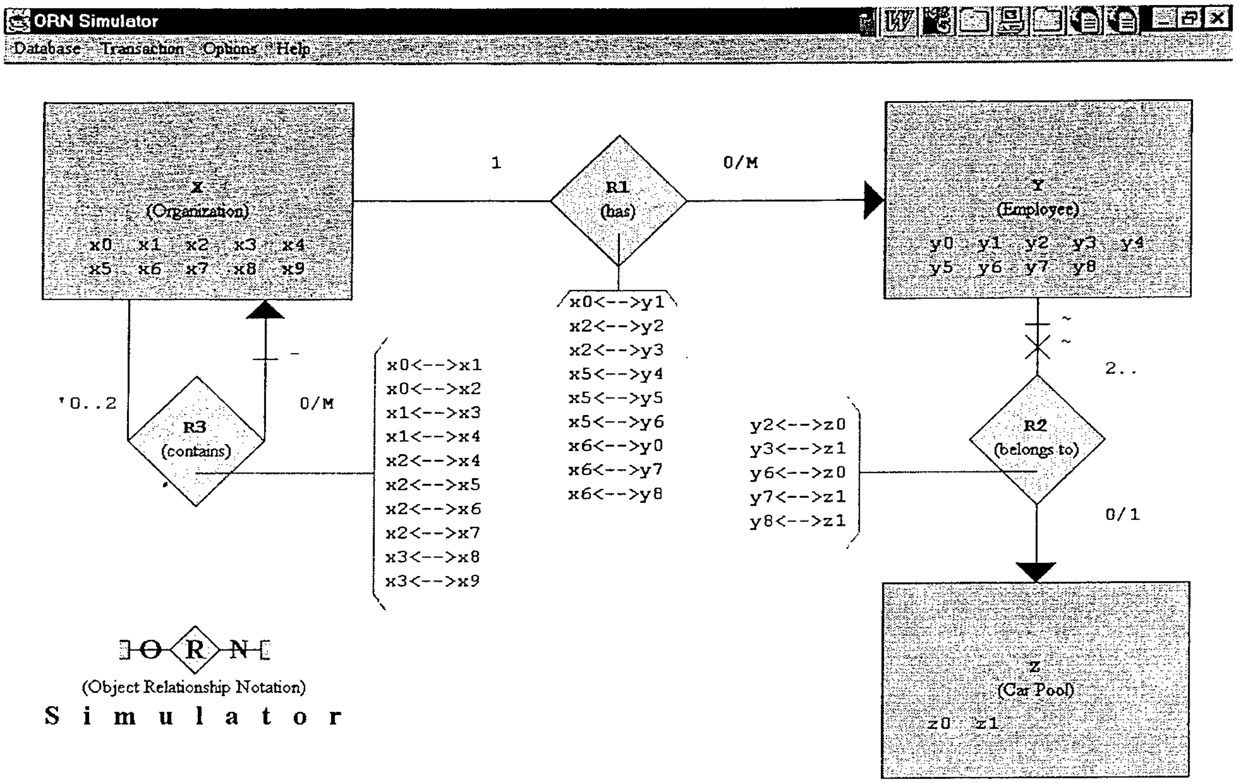
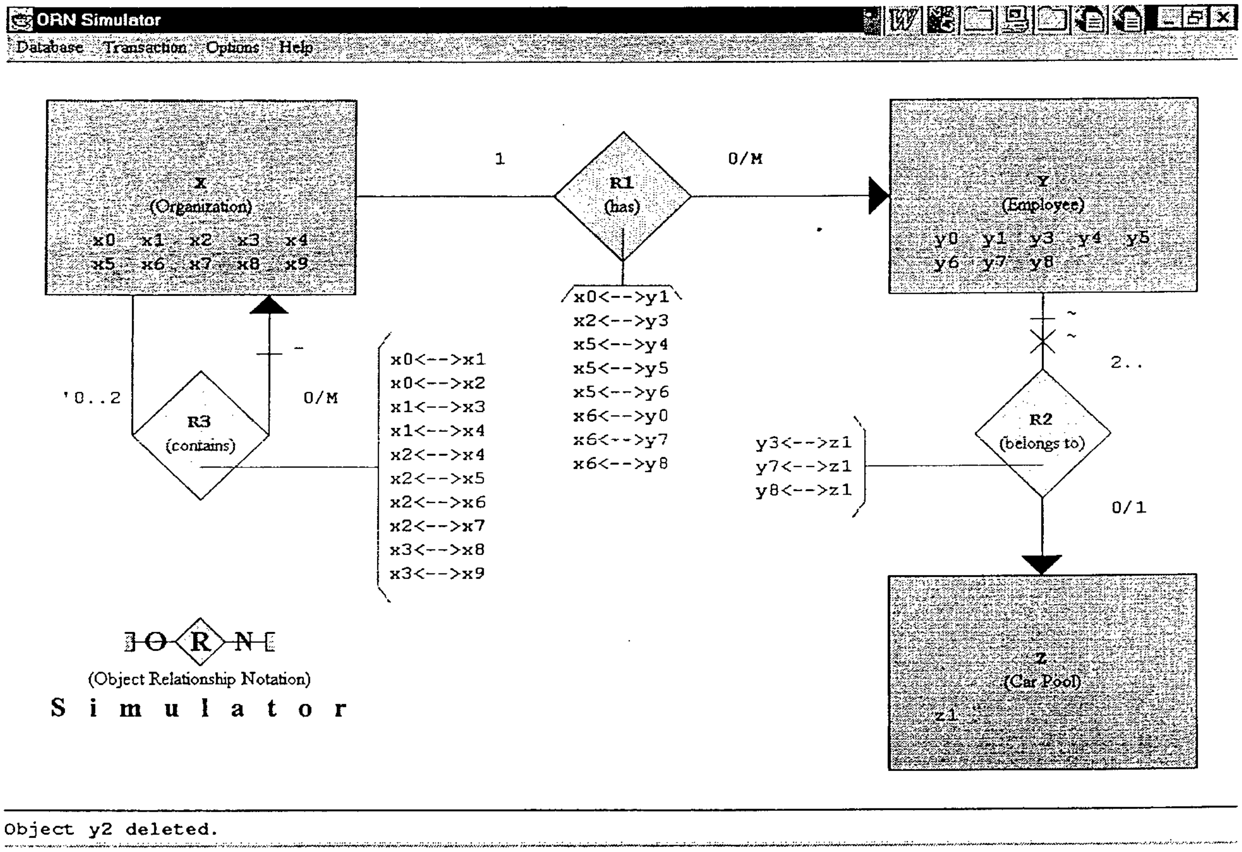


Fig. 3. Deleting Employee y2, One of Two Riders in a Car Pool.



```
Object y2 deleted.
```

```
Database CompanyDB  // Company DataBase
{
  class Employee
  {
    d_String      sSN;  // Soc. Sec. No.
    d_String      name; // Last, First name
    Date          birthDate;
    Org           deptDiv inverse employees
                          <0/M-to-1>;
    CarPool       carPool inverse riders
                          |~X~<2..-to-0/1>;

    ...
  };
  extent Set<employee> employees  key sSN;

  class Org  // Organization
  {
    d_String        name;
    Set<Org>        subOrgs inverse parentOrgs
                            '<0..2-to-0/M>|-;
    Set<Org>        parentOrgs inverse subOrgs;
    Set<Employee>   employees inverse deptDiv;
    ...
  };
  extent Set<Org> organizations  key name;

  class CarPool
  {
    d_String        licNum;  // License Number
    Set<Employee>   riders inverse carPool;
    ...
  };
  extent Set<CarPool> carPools  key licNum;
};
```

Fig. 4. Partial ODDL Specifications

3. Conclusion

The full benefit of ORN is realized when it is supported by a DBMS. Fig. 4 shows how ORN is incorporated into the Object Database Definition Language (ODDL) of OR+ [7]. The partial specifications correspond to the objects and relationships modeled in Fig. 3. Here, *<relationship>*s are associated with the *object-valued attributes* of a class, which implement relationships in an object database—e.g., carPool in class Employee and its inverse attribute riders in CarPool. The Object Database Manipulation Language (ODML) of OR+ provides for database creation, query, and update based on ODDL. The semantics for each *<relationship>* given in ODDL are automatically enforced by the ODBMS. This makes database applications programming less complex and burdensome and increases the level of database integrity.

4. Acknowledgments

This research was supported by NSF grant CDA-9313299 and co-operative agreement HRD-9707076.

References

1. E. Bertino and L. Martino, "Object-Oriented Database Management Systems: Concepts and Issues," *Computer*, IEEE Computer Society Press, 24(4), April 1991, pp. 33-47.
2. S.G. Brown, "The ORN Simulator: A Practical Tool for Modeling Relationship Behavior," student paper (B.K. Ehlmann, advisor), *Proceeding ADMI'97—Increasing Diversity in Research and Education: The Symposium on Computing at Minority Institutions*, ADMI, Washington D.C., May 29 - June 1, 1997, pp. 130-135.
3. P.P. Chen, "The Entity-relationship Model: Towards a Unified View of Data," *ACM Trans. on Database Systems*, 1(1), 1976, pp. 1-36.
4. B.K. Ehlmann and G.A. Riccardi, "A Notation for Describing Aggregate Relationships in an Object-Oriented Data Model," *Applications of Databases—First International Conference Proceedings*, W. Litwin and T. Risch (Eds.), Lecture Notes in Computer Science 819, Springer-Verlag, 1994, pp. 62-77.
5. B.K. Ehlmann and G.A. Riccardi, "An Integrated and Enhanced Methodology for Modeling and Implementing Object Relationships," *Journal of Object-Oriented Programming*, SIGS Publications, 10(2), May, 1997, pp. 47-55.
6. B.K. Ehlmann and G.A. Riccardi, "A Comparison of ORN to Other Declarative Schemes for Specifying Relationship Semantics," *Info. and Software Technology*, Elsevier Science, 38(7), July 1996, pp. 455-465.
7. G.A. Riccardi, "Object Relator *Plus*: A Practical Tool for Developing Enhanced Object Databases," *13th Int'll Conference on Data Engineering*, Birmingham, England, April 7-11, 1997, pp. 412-421.
8 B.K. Ehlmann and M.A. Stewart, "Incorporating Object Relationship Notation (ORN) into SQL," *Proc. 35th ACM Southeast Conference*, Murfreesboro, TN, April 2-4, 1997, pp. 282-289.
9 S.K.Hardeman, "Relationship Behavior in Object Databases: Subtleties and Inconsistencies," student paper (B.K. Ehlmann, advisor), *Proc. of the 34th ACM Southeast Conference*, Tuskegee, AL, 1996, pp. 224-229.
10. W. Kim, "Object-Oriented Databases: Definition and Research Directions," *IEEE Trans. on Knowledge and Data Eng.*, 2(3), Sept. 1990, 327-341.

Applications of Remote Sensing Data*

Martha E.Gutierrez

High Performance Database Research Center
School of Computer Science, Florida International University
University Park, Miami, FL 33199, U.S.A.
(305) 348-1706, Fax (305) 348-1705
http://hpdrc.cs.fiu.edu

ABSTRACT

The NASA Regional Applications Center (RAC) at Florida International University is a division of the High Performance Database Research Center (HPDRC). The RAC Program was initiated by Goddard Space Flight Center's (GSFC) Applied Information Sciences Branch, Code 935, to extend the benefits of its information technology research and cost-effective system development to a broader user community. Timely access to remote sensed satellite data products along with applications designed to meet the needs of the community would promote the wider use of Earth remote sensed data by the RAC regional community and affiliates.

This paper describes the current efforts of the RAC in identifying remote sensed data applications specific to our regional community. First, a brief history of remote sensing will be discussed, and then the different data sets and some of the applications currently being used at the RAC will be presented.

Introduction

As we approach the next millenium, monitoring the Earth's resources and environment has become integral to the future of our planet and mankind. `Remote sensed data is essential to the goal of understanding our Earth, evaluating our resources and ensuring that we do everything in our power to maintain the beauty and richness of our planet.

Remote sensed data are collected by sensors that measure a range of wavelengths of electromagnetic energy reflected or emitted from the Earth. The wavelength bands are often from those parts of the electromagnetic spectrum outside the range of human eyesight, showing normally invisible land characteristics. The data are usually transmitted to Earth where they are processed by computers and archived. Various applications have been developed by Earth Science Experts to evaluate or monitor a variety of conditions on and around Earth. At the NASA Regional Applications Center, we work with remote sensed data toward the goal of facilitating access of this data to the general public. This paper will give an overview of remote sensing history and introduce the variety of remote sensed data available at the RAC and some of their applications.

Overview of Remote Sensing History

Some may argue that remote sensing, as we know it today, began when Galileo used the telescope, in 1609, to peer into space. From the beginning of time, however, mankind has been trying to see the unseen, to look into the depths of the oceans and beyond the stars above. In the mid 1800's, the concept of remote sensing changed when balloonists began taking aerial photographs for the purpose of land surveys and even reconnaissance during the Civil War. The most novel idea was perhaps the Pigeon Corps in 1903, where light cameras which were set to take pictures every 30 seconds, were attached to carrier pigeons. These carrier pigeons then flew to their home shelters obtaining aerial photographs on the way.

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

In 1908 the first aerial photographs were taken from an airplane and soon proved to be a useful tool to make land surveys and maps. In October 4, 1957, the first man-made satellite called Sputnik was launched by the Soviets, which triggered a space race that led to great achievements in remote sensing.

Earth-Probe TOMS Data

The Total Ozone Mapping Spectrometer (TOMS), aboard the Earth Probe satellite which was launched on July 1996, generates the TOMS data. TOMS Earth Probe is part of NASA's mission to planet earth, a long term coordinated effort to study the earth as a global environmental system. In the image of the South Pole, seen on Figure 1 the "ozone hole" is easily identifiable. The dark circle over the North Pole is an area where the satellite failed to record information.

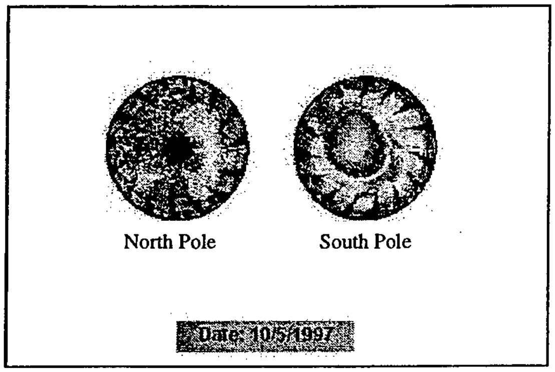


Figure 1 Earth Probe TOMS data

TOMS monitors ozone by measuring ultraviolet light. It has 6 bands in the ultraviolet region of the spectrum. TOMS data has been used extensively to map the ozone hole and is also used to measure sulfur dioxide released in volcanic eruptions.

Landsat TM Data

The Thematic Mapper data is generated from the Thematic Mapper instrument, which has 7 sensors ranging from the Visible to the Thermal Infrared with 30 meter spectral resolution, aboard the Landsat 5 satellite. The Landsat mission is to provide for repetitive acquisition of high resolution multispectral data of the earth's surface on a global basis [1].

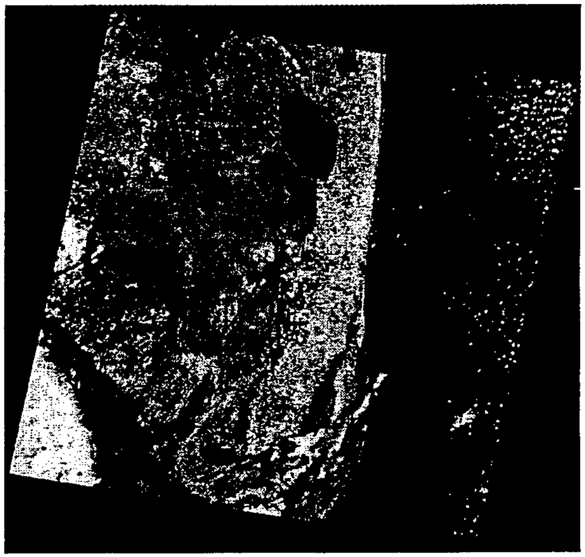


Figure 2 Color Composite Image using sensors 3,2,1 of a South Florida Landsat scene acquired on March 21, 1996.

Landsat TM data has a multitude of applications. These include: cartography, nautical charts of shallow seas, land use, identification of faults and lineaments, geology, mineral prospecting, hydrology, flooding, erosion, wetland environments, agriculture, cropland assessment, crop disease, forestry assessment, forest fires, land use monitoring, urban growth, air pollution, water pollution, storm damage and oceanography. Some of these features can be obtained from selecting the appropriate three bands for the application and creating a color composite image. Figure 2 shows a color composite image of a Landsat scene over South Florida.

Aerial Photography Data

Aerial photography data is a result of the national aerial photography program that was established to coordinate the collection of aerial photographs of the 48 conterminous states every 5 years. This data is acquired as color infrared photography from planes. Flights are 20,000 feet above mean terrain and provide 1 meter resolution images, including threè bands ranging from green to IR. The primary use of color infrared photography is for vegetation studies. Healthy green vegetation is a very strong reflector of infrared radiation and appears bright red on color infrared photographs.

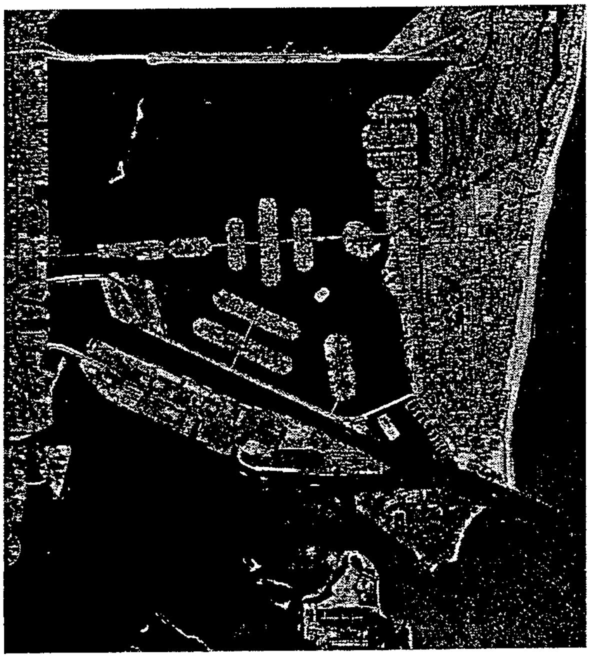


Figure 3 Aerial Photography of Miami Beach, Florida

Applications of Aerial Photography included land-use planning and mapping, geologic mapping, and GIS integration. With GIS systems it is possible to use Aerial Photography as a raster layer to be overlaid with vector data such as pipe and transportation lines, hydrography, public land surveys, roads and trails, and railroads available in Digital Line Graphs (DLG).

GOES-8 Imager Data

The NOAA GOES-8, Geostationary Operational Environmental Satellite, launched on April 27, 1994, is part of the GOES program who's primary mission is the continuous and reliable collection of environmental data in support of weather forecasting and related services. Spacecraft and ground systems work together to accomplish this mission. Of the two sensors aboard, imager and sounder, we are currently ingesting imager data.

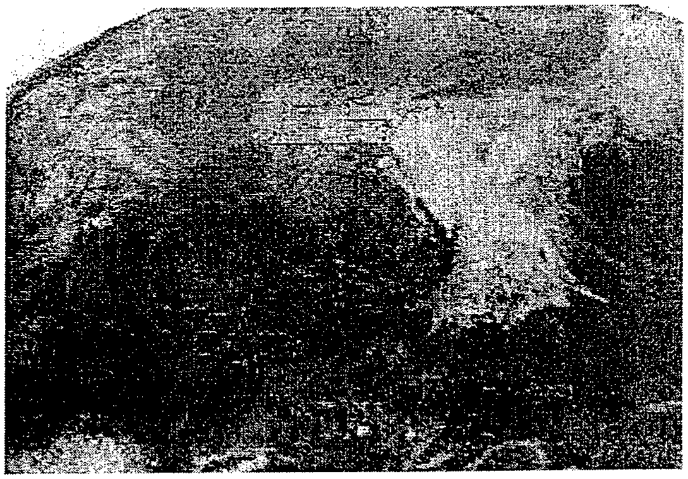


Figure 4 GOES 8 Imager weather data of Eastern US acquired on February 4, 1998

Specific applications of the imager data include severe storm detection monitoring, and tracking; wind measurements from cloud motion; sea surface thermal; precipitation estimates; frost monitoring; rescue operations; and research. The Imager Sensor has 5 bands ranging from the Visible to the Infrared.

SeaWiFS Data

The SeaWiFS Data is generated from the Sea-viewing Wide Field-of-view Sensor launched on the Orb-View 2 (formerly SeaStar) satellite on August 1,1997. The Purpose of SeaWiFS data is to examine oceanic factors that affect global change and to assess the oceans' role in the global carbon cycle. SeaWiFS is a crucial component in a continuing series of comprehensive observations of the ocean in the visible and near infrared that are required for investigations of the marine biosphere. SeaWiFS sensor has 8 bands ranging from the Visible to the near Infrared.

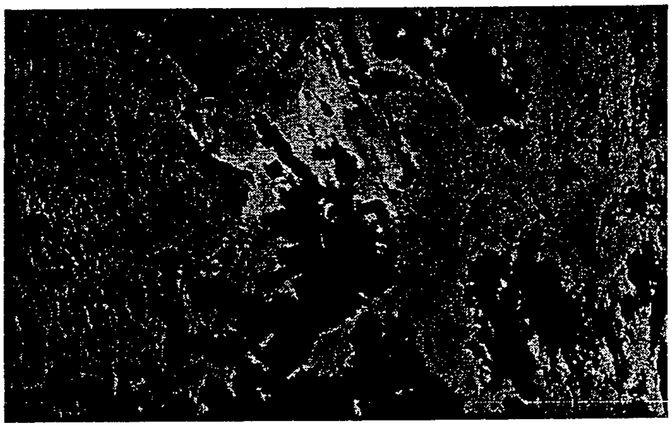


Figure 5 SeaWiFS data shows high concentrations of Chlorophyll *a* surrounding the Galapagos Islands

The image shown in Figure 5 encompasses the Galapagos Islands and surrounding oceans. The chlorophyll *a* algorithm was applied to this image and results show the higher concentration of chlorophyll *a* (the lighter area in the center of the image) surrounding the Galapagos Islands, as compared to the water which is further away from the islands. With SeaWiFS data and correlating data from ships, scientists will, for the first time, have a complete suite of measurements necessary to study seasonal oceanic phytoplankton (unicellular microscopic marine plants). This is done by measuring chlorophyll *a*, which is regarded universally as the most appropriate measure of viable phytoplankton biomass. SeaWiFS can detect dissolved organic material, and suspend sediments from rivers and lagoons are also made possible with. SeaWiFS provides a more accurate prediction of the global climate to changes such as carbon dioxide. Other applications include carbon cycle, sulfur, nitrogen, and ocean influences on the physical climate, including heat storage in the upper ocean and marine aerosol formation.

AVHRR Data

AVHRR data is generated from the Advanced Very High Resolution Radiometer launched aboard the NOAA 12 & 14 Satellites on May 1991 and December 1994, respectively. The National Oceanographic and Atmospheric Administration satellites were originally designed to provide cloud cover information several times per day in an operational environment for meteorological applications. The NOAA Series of satellites have been operating since 1979.

The AVHRR sensor has 5 bands ranging from the visible to the IR. AVHRR data has many applications such as water surface temperature mapping, snow cover mapping, flood monitoring, vegetation mapping, regional soil moisture analysis, wildfire fuel mapping, fuel detection, dust and sandstorm monitoring, observation of volcanic eruptions, mapping of regional drainage, and physiographic features. The normalized difference vegetation index (NDVI) algorithm has been used extensively for large area mapping [2]. Figure 6 below shows NDVI applied to AVHRR data for the state of Florida. The images were acquired during the winter and summer of 1996. The darker color shows the areas with more vegetation. NDVI can be applied to Landsat data as well, resulting in the smaller area mapping with higher resolution.

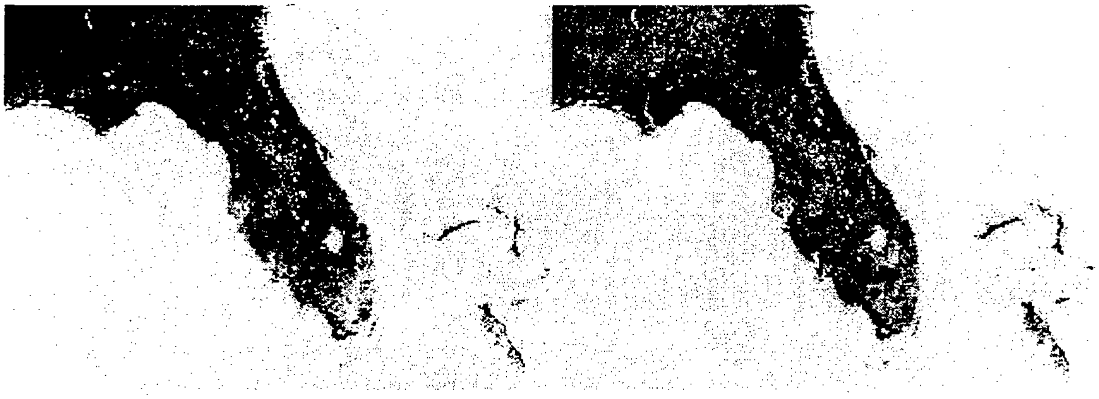


Figure 6 NDVI applied to AVHRR data of Florida - Summer 1996 (left) and winter 1996 (right)

The Future of Remote Sensing

There have been great advances in the technology of remote sensing, in the last 30 years. The future appears equally bright. As part of NASA's Mission to Planet Earth remote sensing activities, the Unmanned Aerial Vehicle (UAV) Project has built a new vehicle to carry remote sensing into the next century. The UAV named "Freewing" will serve as a platform for several remote sensing instruments as well as a test bed for the latest technologies in miniature intelligent avionics, communications and instrumentation. Among the instruments that this vehicle will carry is a Computerized Component Variable Interference Filter Imaging Spectrometer (C2VIFIS). This hyperspectral instrument is capable of acquiring 96 bands simultaneously in the spectral range of 419nm to 860nm. This provides a broader range of applications and more precise results because of the many bands available. This system provides greater flexibility in the temporal and spatial resolution of the data because the client can control the frequency of the flights and the altitude at which the plane flies. The restrictions previously imposed by remote sensing satellites, given that each sustain a unique orbital altitude from the time they are launched to the time they are decommissioned, disappear. Growers, for example, who wish to monitor their crops weekly, can do so, at whatever resolution they wish. Commercial applications include: management and monitoring of coastal zones, wetlands, forests, agriculture, urban planning, municipal zoning and management, reef assessment, pollution monitoring, and low cost field testing of new remote sensing instruments.

Bibliography

[1] Jensen, John R., *Introductory Digital Image Processing: A Remote Sensing Perspective,* Prentice-Hall, Inc, 1996.

[2] Lillesand, Thomas M., Kiefer, Ralph W., *Remote Sensing and Image Interpretation* Third Edition, John Wiley and Sons, Inc, 1994.

GIS and Internet Access to Spatial Data*

Maria C. Martinez

High Performance Database Research Center
School of Computer Science, Florida International University
University Park, Miami, FL 33199, U.S.A.
(305)348-1706, Fax (305)348-1705
hpdrc@cs.fiu.edu , http://hpdrc.cs.fiu.edu

ABSTRACT

Geographic Information Systems (GIS) provide useful information used by many in planning and analyzing studies of Earth resources. These systems consist of geographical databases and software/hardware units to view and study the data. One such way to design and use such a system is via the Internet. A method for accessing GIS databases through the Internet has been developed at Florida International University's High Performance Database Research Center. Algorithms can be performed on the geographic data and accessed through the Internet. Many users can access the GIS system without having to have the data stored locally.

NASA has established at Florida International University a Regional Applications Center (RAC), as part of the High Performance Research Center, to develop applications of these GIS systems for regional institutions. The Internet based applications will provide access to spatial and geographic data to the customers of the RAC.

1.0 INTRODUCTION

Geographic Information Systems integrate four major components: hardware, software, data, and people. Hardware is the computer or network of computers on which the GIS operates. The software provides the tools to store, retrieve, analyze and display the information. These tools should include database management systems (DBMS) and graphical user interface (GUI) for easy usage of the data. The data is geographic data and related tabular data that is managed and maintained by the DBMS. The people or organization is what manages the system and applies it to real world problems.

With the increase availability of computers the Internet is becoming a convenient source of information transfer. Integrating GIS and access to these systems via the Internet facilitate and expands their usage. The database can be stored in a central location and accessed by many users in different locations. What is needed is a friendly easy to understand GUI. Some of the major languages used to create our applications for the Web are: Hypertext Markup Languages (HTML), Java, and Virtual Reality Modeling Language (VRML) that is used for 3D and animation.

Storing the data and retrieving is a major component of the GIS. In the 1970's and 1980's, the database community lumped every kind of data other than fixed-format records (including spatial data) into a heterogeneous group called 'non-standard data'. It was tempting to extend relational database technology, with this simple conceptual structure, to handle all kinds of data. However, relational data is not just a way to represent data, it also implies or suggests certain access algorithms that are particularly efficient on data naturally represented by rows and columns. If we force spatial data into tabular form, for example by introducing relations like faces, edges and vertices may have harmful consequences. Geometric proximity is not reflected by proximity in memory. For example, all vertices no matter how far apart in space are stored contiguously in the same relation, whereas a vertex and its incident edges and faces are scattered all over storage. This may have grave consequences when data is stored on disk, where instead of accessing one

** This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

entire object as a unit we may have to gather bits and pieces of this object in many separate disk accesses [1]

2.0 HIGH PERFORMANCE DATABASE RESEARCH CENTER

The High Performance Database Research Center (HPDRC) is a division of Florida International University (FIU), School of Computer Science. It conducts research on database management systems and various applications, leading to the development of new types of DBMS, new database techniques, and the refinement of existing ones. HPDRC's largest project is the development of algorithms and a prototype of a massively paralleled Semantic/ Object Oriented DBMS, Sem-ODB. Our system is useful for most typical database applications, as well as for specialized domains such as Earth Sciences.

2.1 NASA REGIONAL APPLICATIONS CENTER

The NASA Regional Applications Center (RAC) at FIU is a subdivision of HPDRC. NASA has established approximately 15 RAC's across the country at this time. The RAC Program was initiated by NASA Goddard Space Flight Center's (GSFC) Applied Information Sciences Branch, Code 935, to extend the benefits of its information technology research and cost-effective system development to a broader user community [4]. The RAC objectives are based on the goal of fostering the use of environmental and Earth resource data by regional institutions. The ultimate goal of the RAC is to establish a fundamental set of remote sensing technologies that can be assembled by a specific user community, to meet the information needs of that community.

With this RAC system, NASA will refine and transfer its technology through collaborative test bedding. It will use the RAC created in-situ and ancillary databases to support the calibration and validation of its satellite data. NASA will incorporate the RAC's applied research results into shareable global environmental knowledge databases.

In August 1996, GSFC and FIU established a Regional Applications Center at HPDRC, to expand the practical applications of NASA satellite sensor readings, combined with other physical or logical data to the benefit of the Southeast US region. This strengthen an existing project between NASA GSFC and FIU's High Performance Database Research Center, for the purpose of developing and implementing an advanced new database technology, based on the semantic database model.

The RAC System hardware at FIU currently includes a Goes 8 GVAR Receiver, an Ingest Machine HP Vectra and a HP 9000 Workstation that has the RAC software. The RAC Software has been installed at the FIU RAC in March 1998, and it is the first Version 0 installed in any RAC. The software consist of a Curator unit which does the installation of the ingest algorithms. An Ingest unit does reformatting, calibration, navigation, meta file generation and gif file generation. The Database currently used is Object Store but is in the process of being changed to HPDRC's Sem-ODB. A Planner unit does the scheduling and dispatching and is the control unit.

As a goal of the RAC, there has been established an affiliate program, or prospective customers. These consist of national parks, governmental institutions, academic units, utilities, and agriculture industry. These affiliates will be the ones to use the RAC as a resource to develop applications for their GIS systems.

3.0 GEOGRAPHIC INFORMATION SYSTEM

A Geographic Information System (GIS) is a system of hardware, software, data, people, organizations, and institutional arrangements for collecting, storing, analyzing, and disseminating information about areas of the earth. These systems take satellite data, aerial photographs, digital maps, tabular information and other digital data and process it to make it into some form of report, or map that is meaningful to the end user. A GIS system can be viewed as a process rather than a thing that supports data collection, analysis, and decision making and is more than a software or hardware product. This makes it valuable to public and private institutions in explaining events, predicting outcomes, and planning strategies.

3.1 GIS DATABASE

One of the most important components of a GIS is the data. A database for a GIS system can contain many forms of data. At HPDRC we have acquired many of the following forms of data for use in a GIS database. Digital Line Graphs (DLG) supply users with the digital version of information printed on United States Geological Survey (USGS) topographical quadrangle maps. Digital Elevation Models (DEM) is a data exchange format developed by the USGS for geographical and topographical data. Land Base and Land Cover Classifications are the natural and man-made environmental features with which infrastructure is developed or by which natural resources are indexed and analyzed and are among some classification of classes of structures that can also be stored in a GIS database. TIGER Files, topologically integrated geographic encoding and referencing file, is a type of digital map developed by the United States Bureau of Census to support the 1990- population census. Spatial data or images acquired from remote sensed instruments are another form of data used in GIS systems.

At HPDRC we have also acquired spatial data of our region. We have Earth Probe TOMS data of ozone information, Landsat Thematic Mapper data, Aerial Photography of Dade County, Orb-View 2 SeaWiFS data, Goes 8 Imager data that we ingest from our GVAR ingest system, and NOAA 12 and 14 AVHRR data. All our databases are stored into a Sem-ODB database that is described below.

4.0 SEMANTIC DATABASE

The HPDRC's Semantic DBMS, Sem-ODB, is based on the Semantic Binary Model. In the Semantic Binary Model, the information is represented by logical associations (relations) between pairs of objects and by the classification of objects into categories. The Semantic Binary Model is the most natural and convenient way of specifying the logical structure of information and for defining the concepts of an application's world. It is represented in the form of a semantic binary schema [3]. It stores spatial data in an efficient manner and allows storage of raster, vector and attribute data.

4.1 DESCRIPTION

The Semantic Database models are potentially more efficient than the conventional models for two main reasons. The first is that all the physical aspects of the representation of information by data are invisible to the user and the second is that the system knows more about the meaning of the user's data and about the meaningful connections between such data. The first reason creates a potential for optimization by allowing more changes without affecting the user programs. The second allows this knowledge to be utilized to organize the data so that meaningful operations can be performed faster at the expense of less meaningful operations [3].

The semantic database is perceived by its users as a set of facts about objects. These facts can state that the objects belong to a category, they can state that there is a relationship between objects or they can be fact relating objects to data, such as numbers, texts, dates, images, etc [3]. HPDRC's Semantic DBMS contains semantic facts and inverted semantic facts. This fact inversion scheme assures efficiency of queries including range queries and content access and also exhibits low entropy of data blocks, which facilitates compression.

The mathematical abstraction of the relational model has allowed the introduction of powerful and easy -to-use languages for retrieval and updates of databases. The semantic model however, offers a higher degree of abstraction, which results in more concise user programs, speedier processing (due to optimization), and a wealth of other features. Relational databases are good for general conventional database applications. However, in situations where the structure of information is complex, or where greater flexibility is required (objects with unknown identifiers, or objects moving from one category to another, etc.), or where non-conventional data is involved (spatial data, long text, images, etc.), semantic databases need to be considered.

4.2 LANDSAT SEMANTIC DATABASE SCHEMA

We used a Semantic Binary Database for the storage of GIS data. The first step involved in creating the database is the design of the schema. HPDRC has acquired some Landsat TM data of scenes and quads observed by Landsat 4 and 5. These spatial data along with its meta-data are integrated to the database by the schema design. Fig. 1 shows the current schema design for the Landsat TM database.

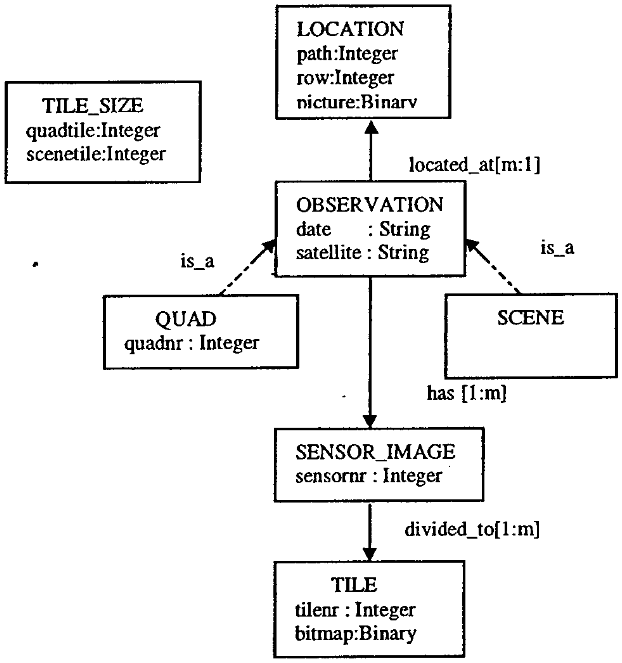


Fig. 1 Schema for Landsat Thematic Mapper database

Description of schema

LOCATION—category (A catalog of locations defined by path and row which is the coordinate system used for Landsat observations)

OBSERVATION—category (A catalog of observations which refers to observations made by a Landsat satellite for a particular date)

QUAD—category (A catalog of quads which is a particular area of observation specified by the quadnr)

SCENE—category (A catalog of scenes which is the area of observation)

SENSOR_IMAGE—category (A catalog of images observed by a sensor on-board Landsat satellite sensor specifies the sensor which made the observation)

TILE—category (A catalog of tiles which are segments of a sensor image)

TILE_SIZE—category (A category of TILE_SIZE which contains the sizes of quad and scene tiles)

located_at –relation from OBSERVATION to LOCATION (m:1,total) (An observation must have a location that it observes. There are many observations with the same location)

divided_to–relation from SENSOR_IMAGE to TILE (1:m) (A sensor image is divided in to smaller segments called tiles)

has–relation from OBSERVATION to SENSOR_IMAGE (1:m) (An OBSERVATION consists of many sensor images observed by different sensors in the satellite)

path – attribute of LOCATION of type Integer (The path number of the coordinate system)

row – attribute of LOCATION of type Integer (The row number of the coordinate system)

picture – attribute of LOCATION of type Binary (The image depicting the region covered by scene)

date – attribute of OBSERVATION of type String (The date when the observation was made)

satellite – attribute of OBSERVATION of type String (The name of satellite which made the observation)

quadnr – attribute of QUAD of type Integer (The quad number which specifies the area of observation)

sensornr – attribute of SENSOR_IMAGE of type Integer (The sensor number which specifies the sensor that observed the image)

tilenr – attribute of TILE of type Integer (The tile number which identifies the tile)

bitmap – attribute of TILE of type Binary (The binary data observed by the sensor on-board the satellite)

5.0 INTERNET ACCESS TO SPATIAL DATA

In the 1990's the Internet has experienced explosive growth. For most of its existence the Internet has been a research and academic network, but as more users become connected in more countries across the world new commercial applications are being used. Commercial enterprises and consumers of different types are recognizing the Internet's potential. People and businesses can now use the Internet to retrieve information and communicate and conduct business, and access services and resources on-line.

To access our GIS database by the Internet we use a client/server model. The client is the interface from the user to the server. This can be implemented in Java, HTML, or VRML for 3 dimensional viewing. The server runs on a host computer and when a client access information it retrieves it from the database and sends it to the client.

5.1 DESIGN OF THE APPLICATIONS

The overall structure of the application consists of design and implementation of 3 main components:

- The design and implementation of a storage-retrieval medium for the images. In our case, it is a semantic database, Sem-ODB, developed at HPDRC. This is described in the previous section.
- The design and implementation of a client program that acts as the front-end of the application. This is a HTML script, Java Applet, or VRML Interface depending on the application design.
- The design and implementation of a server program that queries the database to fulfill the requests of the client.

The client program will interact with the user to compose a query using easy-to-use Graphical User Interface (GUI). It will send the user's request to the server for processing, and will display the results that are received from the server for a particular request. The server program acts as the back-end of the application interacting with the database to fulfill the requests of the clients. This program will query the database to obtain for the client's requests and communicate with the client to send and receive results

5.2 CLIENT PROGRAM

The main task of the client program is to obtain the user's query using an easy-to-use GUI. This can be code written in a combination of ways. It can be done in HyperText Mark-up Language (HTML), JAVA Applets, CGI, or VRML interfaces.

In one example to retrieve Landsat TM Data, the client shows a map like Fig. 1(a) (map of US) from which the user select a state (for example Florida, see figure 1(b)).

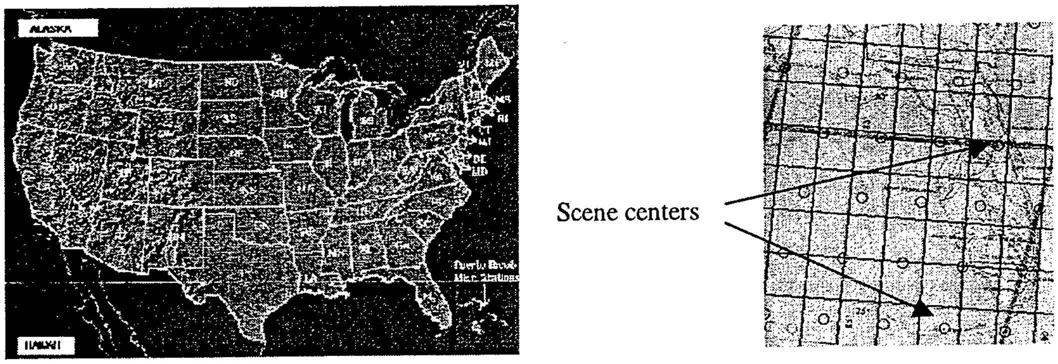


Fig. 1 (a) Map of the US from which the user selects a state of interest (b) Map of the state of Florida with the scene centers marked by circles

The user in most cases requires only seeing a small area of this region. The map of Florida is marked with scene centers. The user selects a scene center of interest, which gives the client program a particular path and row number that the user requires. This information is transmitted to the server program as Query1. The results of Query1, which are received by the client program, contains meta-data on all the Landsat Thematic Mapper images present in the database for the selected path and row. Also, a name of a picture file, which contains the region covered by the scene for the selected path and row, are received. The picture file depicting the selected region is displayed from which the user selects a smaller region of interest.

The client program then calculates the tiles for quads and scene involved in the selected region. Once the tiles required by the user's selection are calculated, it is checked whether the tiles are present in the database. Since the results of Query1 contain all meta-data of Landsat images present in the database for the particular path and row, this can be easily performed. Next, the dates available for the selected region are displayed from which the user selects a date of the observation he/she prefers to view the image. Next, the sensor images available are computed and displayed for the user to select to produce color-composite images. Note that we require sensor images to be computed because the user's selection may span across multiple quads and it is possible that the database may not have sensor images of a quad selected. This is resolved by finding the intersection of sensor images present in database for the user's selected quads. Also, image enhancement could be done to the resultant image by applying a filter. Next the selected tiles, date, sensor numbers and filters are composed into a query (Query2) and transmitted to the server program to process. Finally, the result of Query2 is an image created by applying the selected sensors to red, green and blue and the selected filter. The client program will display this image.

The client program is implemented as a JAVA applet running on a WWW browser. The states (e.g. florida.html) where the user clicks on a preferred state to view an image, is implemented in HyperText Mark-up Language (HTML) with path and row numbers embedded as parameters. The connection between the client and the server is handled by a TCP/IP (reliable byte stream) connection using socket implementation.

5.3 SERVER PROGRAM

The main method or controlling body of the server program provides two major functions. It opens the database and waits for a client to contact. When a client does make a connection, it creates a thread or process and lets the client communicate with the new process for its future transactions as it continues to wait for more clients.

In our example the client requests for two different types of queries from the server.

1. Query1: For a given path and row number, provide meta-data on all the Landsat quads and scenes present in the database for the particular location along with a picture of the region.
2. Query2: For a given path, row and sensor numbers, query the bitmaps for the selected tiles.

The server waits for query from the client. On receiving a request, checks whether it is of type Query1 or Query2 and perform the necessary tasks accordingly. It then sends the results for the queries to the client and waits for another query. The server program is implemented in C++. It uses the C++ interface developed for the Semantic Binary Database at HPDRC to query the database. The server runs on a Sun Sparc station using Solaris as the underlying Operating System. Note that the server side is kept simple intentionally so as to make the application easily portable between different schemas of the database. The main components that require to be modified are the two queries.

6.0 CONCLUSION

The Internet has become a tool of information transfer that is used by many public and private institutions. It is feasible and desirable to access Geographic Information Systems through the Web. This has the advantage that the data can be centralized and many applications developed for viewing it. The user does not have to have a GIS database it is kept by the application center.

The applications can perform algorithms on the data before sending it to the user to view. Tasks such as geolocating, or applying filters can be added to the model. In our example there is the capability of producing color-composite Landsat images by applying any of the different sensors images to the RGB color model. Applying filters can further enhance these resultant images.

7.0 REFERENCES

[1] J. Nievergelt, M. Freeston, "Special Issue Editorial Spatial Data: applications, concepts, techniques", *The Computer Journal*, Vol. 37, No. 1, 1994.

[2] N. Rishe, Accepted grant proposal to NASA. Grant NAGW-4080. 1992a.

[3] N. Rishe, *Database Design: The Semantic Modeling Approach*, McGraw-Hill, 1992b.

[4] Applied Information Sciences Branch/Code 925, NASA Goddard Space Flight Center, White Paper 1997.

Knowledge-Based Image Retrieval with Spatial and Temporal Constructs

Wesley W. Chu

Computer Science Department
University of California, Los Angeles
wwc@cs.ucla.edu, http://www.kmed.cs.ucla.edu

ABSTRACT

A knowledge-based approach to retrieve medical images by feature and content with spatial and temporal constructs is developed. Selected objects of interest in a medical image (e.g. x-ray, MR image) are segmented, and contours are generated from these objects. Features (e.g. shape, size, texture) and content (e.g. spatial relationships among objects) are extracted and stored in a feature database. Knowledge about image features can be expressed as a hierarchical structure called a Type Abstraction Hierarchy (TAH) which is user- and context- sensitive. Knowledge based query processing that provides approximate (e.g. similar to, near to, etc) matching of image features and content are developed. Further, a visual query language has been developed that accepts visual iconic input on the screen. User models are introduced to provide default parameter values for specifying query conditions. We have implemented a Knowledge-Based Medical Images Database System (KMeD) using the above mentioned technology at UCLA. The results from this research should be applicable to other multimedia information systems as well.

Selected Publications

1. Chu, W. W., I. T. Ieong, and R. K. Taira, "A Semantic Modeling Approach for Image Retrieval by Content", special issue on Spatial Database Systems, Journal of VLDB, 3(4), 1994, pp. 445-477.

2. Chu, W. W., A. F. Cardenas, R. K. Taira, "KMeD: A Knowledge-Based Multimedia Medical Distributed Database System", special issue on Scientific Database Systems, Journal of Information Systems, June, 1995.

3. Chu, W. W., K. Chiang, C. C. Hsu, and H. Yau, "An Error-Based Conceptual Clustering Method for Providing Approximate Query Answers", in Communications of the ACM, Vol. 39, No. 13, December 1996, http://www.acm.org/pubs/cacm/extension

4. Chu, W. W., C. C. Hsu, I. T. Ieong, and R. K. Taira, "Content-Based Image Retrieval Using Metadata and Relaxation Techniques", in Managing Multimedia Data: Using Metadata to Integrate and Apply Digital Data, edited by Wolfgang Klas and Amit Sheth, McGraw Hill, 1998.

5. Chu, W. W., I. T. Ieong, R. K. Taira, and C. M. Breant, "A Temporal Evolutionary Object-Oriented Data Model and its Query Language for Medical Image Management", in Proceedings of the 18th International Conference on Very Large Databases, pp. 53-64, August 1992.

Semantic Schema Editor and Viewer Tool for Accessing, Designing and Interacting With Semantic Data Base*

Yaman Battikhi, and Maria Martinez

High Performance Database Research Center
School of Computer Science, Florida International University
University Park, ECS251, Miami, FL 33199
(305) 348-1706, Fax (305) 348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

Data from remote sensing is becoming very important in management and protection of forestry and agricultural land. Over the years the remotely sensed data collected has grown exponentially, necessitating informational processing technologies to store and retrieve these data efficiently. The High-Performance Database Research (HPDRC) at School of Computer Science, Florida International University has developed the Semantic Binary Model (Sem-ODB) that is efficient in storing and retrieving these types of spatial datasets.

We developed an online graphical tool that interacts with the Sem-ODB. This tool provides an easy-to-use GUI interface that creates and views a Sem-ODB schema diagram. The tool also has the capability of querying the Semantic Database visually using the Schema diagram. Currently, HPDRC has developed Semantic Databases for SeaWiFS, Landsat, Ozone, Ocean Temperature and other geospatial data. The Schema Editor/Viewer tool works as an interface between a World Wide Web user and the Semantic Database to access these data. We hope to advance this tool to pose complex queries using better easy-to-use GUI tools.

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

Interfacing Java to Semantic DBMS*

Dmitriy Beryoza, Jinny Uppal, Philippe Pardo, and Naphtali Rishe

High Performance Database Research Center
School of Computer Science, Florida International University
University Park, Miami, FL 33199, U.S.A.
(305) 348-1706, Fax (305)348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

The Java-to-Sem-ODB Bridge is a Java Native Interface library that provides software designers with a convenient way to access Semantic Database (Sem-ODB) functionality from a Java application. With its help truly portable and Internet-enabled Java applications can be developed easily. At the same time the uniformity of access to Sem-ODB is not sacrificed. Built on top of the standard Sem-ODB function library, the bridge maps Sem-ODB classes, methods and data types into the corresponding Java classes and data types. The bridge also saves Sem-ODB developers the trouble of creating, testing and maintaining a parallel, Java-based version of the Sem-ODB library, therefore saving a considerable amount of time and effort. Versions of the bridge for both Sun Solaris and Windows NT have been developed. The Java-to-Sem-ODB Bridge is currently being used in various HPDRC research projects. A number of applications have been developed using it, including a CORBA-enabled Java interface to Sem-ODB, a web-crawler and an online Internet search engine. The bridge is being extensively tested and an enhanced and optimized version is being developed.

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

Creation of 3D Satellite Imagery*

Debbie Davis

High Performance Database Research Center
School of Computer Science Florida International University
University Park, Miami, FL 33199, U.S.A.
(305) 348-1706, Fax (305) 348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

Satellite imagery is an important tool that has numerous applications. It can be used to gather information on the state of the environment, for commercial purposes and for education. As useful as satellite images have been found to be, however, they are two-dimensional. With today's technology, it is possible to render 3D satellite images that are more visually appealing and useful for data classification. To render a 3D satellite image, a standard U.S.G.S. Digital Elevation Model (DEM) file is combined with corresponding satellite data. DEMs are topographic map data stored as 16-bit gray-scale raster image files where dark tones represent low areas and light tones represent areas of higher elevation. The satellite images are typically Landsat images. Although there are a number of ways to successfully render 3D satellite images, one of the most flexible ways of doing so is through the use of POV-Ray (Persistence of Vision Raytracer). POV-Ray is a 3-dimensional raytracing engine. It uses a scripted based graphics language which takes the supplied information and simulates the way light interacts with objects or images to create 3D pictures. Input files are typically in the form of targa bitmaps. DEM and satellite images are converted to targa bitmaps and included in a POV-Ray script. Within this script, options such as the location and angle of the camera and light source can be manipulated to produce the desired result. This presentation briefly discusses a sample script used to create one of these 3D images and demonstrates the rendering of the corresponding image.

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

The Spatially Enabled Enterprise: Using Object-Relational Database Technology And Open Interfaces To Support Spatial Data Warehouse Applications

James A. Farley

Technical Director
University of Arkansas,
Center for Advanced Spatial Technologies

Recent initiatives across the spectrum of information technology have focused on basic problems of openness, access and published standards as a basis for developing interoperable strategies and products (e.g. Object Management Group). Clearly, the growth of technologies such as the Web, Enterprise Java Beans, eXtensible Markup Language (XML) and others reflect this trend. The proliferation and maturation of these foundation technologies is a tangible response to users and user communities that require maximum access to rich, diverse information and demand minimal constraints associated with the process of locating and consuming data. Leveraging from this trend, the University of Arkansas, Center for Advanced Spatial Technologies, and a team of private sector partners, is assembling a comprehensive, spatially-enabled data warehouse, that will deliver rich, complex, data to users in state and local government, K-12 programs, and a range of other clients throughout the state. The Seamless Warehouse of Arkansas Geodata or SWAG is being constructed using object-relational technology and open, industry sanctioned interfaces which support access from multiple heterogeneous client-side applications. When it is complete SWAG will exceed one terabyte in size and will house metadata, attributes, spatial geometry for vector and raster data and will support data delivery, data mining and data warehousing applications via domain specific, spatial middle-ware.

Western Cultures Semantic Database Application*

Raquel Ghersgorin, Abraham Anzardo, and Naphtali Rishe

High Performance Database Research Center
School of Computer Science, Florida International University
University Park, Miami, FL 33199, U.S.A.
(305) 348-1706, Fax (305)348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

. The High Performance Database Research Center (HPDRC) recently committed to developing a database application for the Florida International University (FIU) President's office. This application is required to store data about persons considered to be historical figures (according to book references collected by the office), due to their contributions in a specific field.

Due to the impact of Semantic modeling, the database was designed and modeled using this methodology. The database is represented by a semantic schema and is currently being developed using the following tools:

1. C++ application to create the database, and load the initial data provided by the user.
2. Use the WWW Database Application Interface Tool (developed at HPDRC and called WebRG) to display and generate reports of the data through a Web interface.
3. HTML input forms to insert and update the semantic database content, and JavaScript language to validate the forms.
4. Use tools for including a graphical representation of the data (statistical charts).

The resulting semantic application will permit the efficient storage and retrieval of data, the useful display of reports and the informative graphical representation of the data through a Web interface. All of these to support research assertions about the impact of historical facts in the evolution of western cultures.

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

Creating A Semantic Database to Hold SeaWiFS Data, That Can Be Used By Scientists Studying Ocean Color*

Alejandro Gonzalez

High Performance Database Research Center
School of Computer Science, Florida International University
University Park, Miami, FL 33199, U.S.A.
(305) 348-1706, Fax (305)348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

At the High Performance Database Research Center, we are developing a database to hold SeaWiFS Data over a period of time so scientists can study the changes of ocean color. Scientists will be able to query the database according to the concentration of phytoplankton, date, latitude and longitude. This database will be growing as the number of applications to SeaWiFS grows. Through the use of JAVA, scientists can preview animated movies of several SeaWiFS images; they will be able to watch as the ocean color changes over a period of time.

Future projects involving SeaWiFS are integrating the SeaWiFS Database to SEADAS and the RAC Software, which are developed by NASA. The major goal is to deliver products to scientists around the world.

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

GOES 8 GVAR Ingest System*

Freddy Haayen

High Performance Database Research Center
School of Computer Science, Florida International University
University Park, Miami, FL 33199, U.S.A.
(305)348-1706, Fax (305)348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

At the NASA Regional Applications Center, at Florida International University, we have set up a ground system to receive and process meteorological data. GOES series of satellites are owned and operated by the National Oceanic and Atmospheric Administration (NOAA). Once the satellites are deployed, NOAA assumes the responsibility for command and control, data acquisition, product generation and distribution. Each satellite in the GOES series carries two major instruments: the IMAGER and the SOUNDER MODULES. These instruments resolve visible, infrared, temperature, and moisture profiles from the atmosphere. They continuously transmit spatial data to ground terminals where it is processed, then re-transmitted to the satellite where we retrieve the newly edited data called GOES GVAR Retransmission Format.

Running under IBM's OS/2 operating System, our Ingesting machine is a combination application software and hardware that is configured on a Hewlett Packard Microcomputer. Once the system is running properly and hooked up to a local satellite dish, the receiver begins to ingest environmental data from either the GOES 8, GOES 9, or GOES 10 satellites in a real time mode.

A fully ingested image can be displayed as a still image in any one of the spectral bands: one visible, or four infrared bands. The collected spatial data files could also be assembled and made to simulate weather loops patterns over the Continental United States or Full Earth Disk view. The resulting image may also be displayed in black and white, or as a false color composite.

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

Integration of a GIS and a Semantic Database System*

Tin Ho, Christian M Pesantes, Venkat Maddineni

High Performance Database Research Center
School of Computer Science, Florida International University
University Park, Miami, FL 33199, U.S.A.
(305)348-1706, Fax (305)348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

There has been a remarkable recent surge of environmental, agricultural, scientific, and academic interest in Geographic Information Systems (GIS) since their graphical nature allows planners to easily visualize the data, which aids in decision making. A GIS is a sophisticated computer based mapping and information retrieval system, consisting of three primary components: a powerful computer graphics program, a set of analysis tools, and one or more external databases. All these components must be tightly integrated; the selection of the right Database Management System (DBMS) plays one of the most important roles in obtaining a high performance and efficient GIS.

An efficient semantic database with an object-oriented framework has been developed at Florida International University's High Performance Database Research Center. In contrast to traditional relational databases, the semantic database provides a much more intuitive graphical schema design as well as better performance in spatial data storage and query processing. GIS generally depend on a database engine to store data. None of the current commercial GIS products use a semantic database for this purpose -- they typically rely on traditional relational databases. We are addressing this limitation by developing an extension that will integrate ArcInfo with our semantic database system.

The integration of this GIS and the Semantic Database can benefit several communities, including agriculture, environmental and forest management, which are currently using GIS systems for their research and studies. The improvement in the efficiency of the GIS would allow the incorporation of a large amount of remotely sensed data into the coverage, since the database is able to handle a large storage and fast retrieval of satellite data as well as textual information about areas. Thus, more complex tasks can be performed.

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

Techniques of Software Quality Assurance at the FIU High Performance Database Research Center*

Dr. Andrei B. Kirienko, N. Rishe, Obando, Huachu Liu, Xiao Long Liu, Ji Quan Lu, Wenhui Meng, Marat Zborovskiy

High Performance Database Research Center
School of Computer Science, Florida International University
University Park, Miami, FL 33199, U.S.A.
(305)348-1706, Fax (305)348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

The Quality Assurance (QA) group was formed in the High Performance Database Research Center (HPDRC) one year ago to test developing software and to inspect its quality. More than ten projects are being developed in the Center. One of tasks of the QA group is to create an automated library of acceptance and regression tests for verification that the software products work properly and satisfy HPDRC standards and NASA requirements. The group members use SQA Suite software to plan, develop, and execute regression, performance and configuration tests for Graphical User Interface (GUI) Windows applications. The QA group uses Software Configuration Management System (Perforce) to control the quality of projects source code. The system facilitates the sharing of files among multiple users and provides version control, release management, defect tracking and build management.

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

Reverse Engineering: Relational to Semantic Database System*

Karthik S. Madhyanapu, Philippe Pardo

High Performance Database Research Center
School of Computer Science, Florida International University
University Park, Miami, FL 33199, U.S.A.
(305)348-1706, Fax (305)348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

The relational to semantic converter is a reverse engineering tool, which given a relational database as input, processes it to obtain an approximation of a semantic database. The conversion is a three-step process, which is completely automated. The tool automates the process of creation of categories, loading objects of those categories, making the relations between these categories and deleting extraneous information present in a relational database.

To demonstrate the applicability of this tool, it has been successfully used to obtain an approximation of a semantic database with data loaded for the Everglades National Park project (ENP). The ENP project is a joint project between Florida International University and the ENP aimed at constructing a set of 22 databases in Oracle, starting from a semantic database design. It has also been integrated with other tools developed at the High Performance Database Research Center (HPDRC).

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

Western Cultures Oracle Database*

Wing Ng

High Performance Database Research Center
School of Computer Science
Florida International University
University Park, Miami, FL 33199, U.S.A.
(305)348-1706, Fax (305)348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

The president database of historical figures has as its purpose, the collection of information about historical figures dated from the beginning of time to the present. It will allow users to do searches, queries about a particular individual, and other functionality, such as printing, graphic representation of the retrieved data, and access of the database from the Internet.

Using "ORACLE" as our tool for implementing this database, we are going to organize and manipulate the information provided by the president's office (ongoing). We will develop a set of tables and queries, based on data and requirement by the President office and load data into the set of tables, which makes up the database. Following the previous, we will develop a set of forms that will display information along with the corresponding interface for displaying data graphical form. Finally, we will make the database available through the World Wide Web.

As a result we would have a database programmed in the Oracle Structured Query Language, capable of providing the user with facilities of performing queries, searches, graphical display of retrieved data using Oracle forms and graphics, and Internet access using Oracle net. This database will serve the purpose of supporting research assertion of the impact of historical figures in the evolution of western cultures.

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

A Priori Selection of Ellipsometry Angles and Wavelengths for In Situ Process Control*

Joe Pontillo

High Performance Database Research Center
School of Computer Science, Florida International University
University Park, Miami, FL 33199, U.S.A.
(305)348-1706, Fax (305)348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

Modern ellipsometers only barely resemble those of a decade ago. In situ measurements of film growth are commonly spectroscopic and the number of measurements for an analysis can be 10,000 or more. Selecting "good" incidence angles and wavelengths is important because at the same time it reduces both measurement time, computational load, and improves the solutions. Currently we choose to say "good" points have high "resolution" (a change in the desired parameter results in a measurable change in measured parameters) and a solvable "condition" (the inverse Hessian matrix is well conditioned for variably damped least squared). We have shown that selection of "good" incidence angles and wavelengths for ex situ ellipsometry can be assisted by systematic evaluation of the resolution and condition of the Hessian. The in situ problem, necessary for process control, adds the dimension of the growing (or shrinking) film thickness. In this work, values of resolution and condition are simulated for about 1000 points in wavelength–angle space.

Visualization of the data allows selection of good angles and identifies wavelengths, which contribute best to the solution of the system of ellipsometry equations. These results are stored in a Sem-ODB high performance semantic database, which can be accessed over the Internet.

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

Landsat Viewer: Create Color Composite Images of Landsat Thematic Mapper Data[*]

Guido Pozo, Eduardo Perez, Rukshan I. Athauda

High Performance Database Research Center
School of Computer Science, Florida International University
University Park, Miami, FL 33199, U.S.A.
(305)348-1706, Fax (305)348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

Data from Landsat Thematic Mapper (TM) sensors detect reflected radiation from the Earth surface in the visible and near-infrared wavelengths. The characteristics of the TM bands can be selected to maximize their capabilities for detecting and monitoring different types of the Earth resources. The ground area covered by one Landsat scene is over 34,000 square kilometers and represents about 260 MB of data.

This demonstration provides a description of a web page interface to a Landsat TM Semantic Database, being developed at the High Performance Database Research Center (HPDRC) at Florida International University. The web interface allows the user to graphically select areas of the Earth to be examined showing latitude and longitude coordinates. The user can further choose the size of the Landsat scene or quad and the color composite image to view based on the seven available sensors. The color composite images are generated in real time in 24-bit color and are subject to various user selected picture enhancement algorithms before being recomposed and exported to the client program in a standard image format. This Landsat Viewer facilitates image processing from the Internet.

[*] This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

Report Generator Web Interfaces to Semantic Database Cover[*]

Naphtali Rishe, Khaled Naboulsi, and Michael Olivero

High Performance Database Research Center
School of Computer Science, Florida International University
University Park, Miami, FL 33199, U.S.A.
(305)348-1706, Fax (305)348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

Report generators are tools that provide an effective way to present data retrieved from databases in a customizable and printable format. Most of the information in a report comes from an underlying table, query, or SQL statement, which is the source of the report's data. Other information in the report is stored in the report's design. Users can manipulate the data, perform necessary mathematical calculations, create charts, and more. Because one has control over the size and appearance of everything on a report, end-user can display the information the way he/she wants to see it. The trend today is to generate reports via Web browsers and many businesses and information providers have regarded the Web as the most powerful and extensible solution for providing up-to-date information for their clients. One way to do this is to link the information power of the database to the Web and to provide a tool to manage all the information needs. Through a Web-based report generator, one can have access to the database in customizable reporting capabilities and to every field in the database. In addition, because the user can have total access to the database, he/she can even make custom database modifications using the report generator. We present the three report generating tools developed at the High Performance Database Research Center (HPDRC) at Florida International University.

The three techniques and tools that provide database connectivity for HTTP Web servers run on Unix and Windows. Each tool offers different ways to construct an SQL query, extract data from the database, and generate HTML pages to produce interactive and real-time Web reports. Moreover, each tool follows a different strategy and has certain characteristics. The first tool, WebRG, allows database integrators to easily develop Web forms and reports for Sem-ODB (Semantic Object Database Management System) and other databases that are ODBC compliant. This tool merges HTML documents with database functions to create a powerful dynamic access to databases using designer-defined macro files. End-users can then query the database through a series of pre-defined forms and reports provided by the application developer. Therefore, users can easily publish data from their databases in the form of Web-enabled reports. Another tool, ***Sem-Access***, allows end-users to have automatically generated forms and standard or customizable reports derived from the conceptual schema of Sem-ODB. This tool is generic in the sense that it provides a simple and effective method to retrieve and manipulate the semantic database and generate reports without requiring pre-defined forms, and to define quickly report content and format. At run time, the Sem-Access extracts data from the semantic database and generates HTML pages to produce interactive and real-time Web reports. Furthermore, the end-user can extract information about the schema of the database itself. This can be enhanced to include whatever information end-users and database developers need to know about the database. A third tool, ***Web-SQL***, is most effective for those users who are familiar with SQL. The user can edit an SQL query, process it, and retrieve results in a tabular format at run-time. These tools are also useful for batch scripts for production of printed reports and for data import/export and post processing.

[*] This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

Aerial Photography Internet Access Tool*

Sangeetha D Sridhar and Alejandro Gonzalez

High Performance Database Research Center
School of Computer Science, Florida International University
University Park, Miami, FL 33199, U.S.A.
(305)348-1706, Fax (305)348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

This program was designed to provide an internet interface between the users and a database that stores the aerial photography data. This program allows you to manipulate and display the aerial photographs. The aerial photographs used here were that of Dade County. The adjacent photographs contained overlapping areas. The overlapping areas were trimmed from each image and then joined to get a virtual image of the whole Dade County. The image was then divided into tiles and stored in a semantic database Sem-ODB for efficient storage and retrieval. The main features are that it allows you to view the schema of the database, browse the information in the database, select a portion of a large image with the mouse by clicking and dragging the mouse until the desired area is covered. This enables the user to view the selected image clearly.

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.

Storage and Visualization of Ozone Layer Thickness Data*

Xiangyu (Grace) Ye and Martha Gutierrez

High Performance Database Research Center
School of Computer Science, Florida International University
University Park, Miami, Fl 33199
(305)348-1706 Fax: (305)348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

The High Performance Database Research Center (HPDRC) at Florida International University (FIU) has been involved in the research, storage, and visualization of several remotely sensed data sets including Ozone Layer Thickness data. This project covers the storage and visualization of the Ozone (Total Ozone Mapping Spectrometer, TOMS) Layer data from three different satellites: Nimbus-7, Meteor-3, and Earth Probe. A Sem-ODB Database has been designed and created. All the textual data including instrument, satellite, frequency and date, as well as the spatial Ozone layer thickness data for about twenty years have been loaded into the database using the Semantic Database Management System's (DBMS) Binary Database Interface that has been developed at the HPDRC.

A friendly graphical user interface has been created together with the main system areas: display process, data manipulation, and data retrieval. All these components are tightly integrated to form a practical interactive system that facilitates the interpretation, manipulation, visualization, analysis, and display of the Ozone data through different platforms including Solaris, Windows, and the Internet. At the same time, during the development of the system, several storage methods and data transfer techniques were tested.

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), AFRL (F30602-98-C-0037), and the State of Florida.